DOCUMENT RESUME

ED 163 965 IR 006 617

TITLE	Association for Educational Data Systems (AEDS). Proceedings of the Annual Convention (16th, May 16-19, 1978, Atlanta, Georgia).
INSTITUTION	Association for Educational Data Systems, Washington, D.C.
PUB DATE	May 78
NOTE	435p.; Not available in hard copy due to marginal reproducibility of original
EDRS PRICE	MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS	Computer Assisted Instruction; Computer Managed Instruction; *Computer Oriented Programs; *Computers; *Conference Reports; Data Bases; Educational Innovation; Educational Technology; *Electronic Data Processing; Information Networks; Professional Associations; Technological Advancement

ABSTRACT

This convention proceedings report examines the latest developments in the application of technology to the educational process. Texts of presentations are included for (1) 15 papers concerning administrative applications in computer based programs, (2) 34 papers elaborating on computer assisted instruction, (3) two papers covering computer assisted guidance, (4) 13 papers on computer assisted testing, (5) two papers exploring computer designed instruction, (6) four papers discussing computer managed instruction, (7) one paper investigating hardware in the behavioral sciences, (8) two papers describing software, and (9) 12 papers of general interest. Thirteen late papers reflecting several of the topics covered are also included. (JEG)

****

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

****

ED163965

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

16th ANNUAL CONVENTION

Association For AEDS *Educational Data Systems*

PROCEEDINGS

May 16-19, 1978

Sheraton-Atlanta Hotel

ATLANTA, GEORGIA

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Shirley Easterwood

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

IR006619

preface

The Association for Educational Data Systems (AEDS) is a private, non-profit educational organization founded in 1962 by a group of professional educators and technical specialists in educational applications of computers and technology. The aim of the organization is to provide a forum for the exchange of ideas and information about the relationship of modern technology to education.

AEDS membership is open to all persons interested in learning more and keeping informed about current developments and directions in educational data systems and computer technology. Included in the membership are educators and technical specialists from all over the United States, Canada, and other countries representing public and private schools, higher education, professional education, state and provinicial departments of education, federal governments, the military, and other professional and technical groups interested in educational use of computers and technology.

This year's Annual Convention, the Association's 16th, is designed to bring together professionals who are interested in learning about and sharing information about the latest developments in the application of technology to the educational process. These proceedings represent a significant contribution to the growing body of knowledge in this vital area. The Convention Committee is grateful to the many contributors to this document. The Committee firmly believes that this collection of papers can have considerable impact upon the educational community if all participants give each paper the careful attention it is due. Additional copies of these proceedings can be obtained for ten dollars ($10.00) from:

The Association for Educational Data Systems
1201 Sixteenth Street, N.W.
Washington, D.C. 20036

The Convention Committee would like to express its appreciation for the support and assistance of all of those who have shown an interest in this year's Convention. The AEDS Board of Directors, the membership, and other interested persons have exhibited interest and initiative which are essential to making the Convention a success. These and others who have contributed financially to the publication of these proceedings are due a note of appreciation and gratitude.

Most especially, the Convention Committee would like to thank those who worked tirelessly to assure that the many details associated with such an undertaking were attended to. The list of those who worked to make this Convention a success are too many to mention - but they know who they are - and their dedication is greatly appreciated.

Thomas J. McConnell, Jr.
Jackie M. Reynolds
Convention Coordinators

James E. Eisele
Program Chairperson

John S. Edwards
Assistant Program Chairperson

table of contents

ADMINISTRATIVE APPLICATIONS

		page
Alverson, Michael	Educational Computer Center Security	3
Avelsgaard, Carol	A Computer System for Financial Aids	6
Birkin, Stanley J.	The Application of MIS Concepts to Course Planning	11
Bryson, Charles H. Smith, Glynton	Data Flow in a Statewide Higher Education Management System: The Need for a Two-Way Street	13
Caskey, Don	One School - One Computer A Secondary School's Mini-Computer Application	18
Cook, James R. Johnson, Marvin A. Gallagher, Michael C.	A Coordinated MIS Approach to Computerized Accounting Systems Design	24
Ewart, Judith C.	ADP Production Technicians: Necessary Personnel for Small DP Centers	28
Fay, Timothy J.	Toward Development of Public Telecommunications Centers	32
Hansen, Tom Klassen, Dan Lindsay, Jim	The Impact of Computer Based Information Systems Upon School District Administration	40
Johnson, Stanford H.	On-Line Registration at East Tennessee State University - Just A First Step Toward an Integrated Database	45
Kros, Frank A. III	Computing Needs of A Community College: Academic and Administrative	49
Marshall, Sr. Patricia	Experiences With A Computer Advisory Council	55
Murphy, John J.	Determination of Data Processing Requirements and Equipment in A Moderate Size State University	60
Powers, Michael J.	A Management Information System for Academic Administrators	66
Raucher, S. M.	Data Security & Privacy: Implications for Education	71

COMPUTER-ASSISTED INSTRUCTION

Adkins, Kathleen	Success Story: CAI in the Elementary School	76
Allee, John G. Quinn, Peter J. Williams, Robert L.	English Usage Exercises: A CAI Series	80
Arneson, John Jamison, Robert	Four Applications Relating Teacher Needs to A Timeshare System	83
Barron, Jonathan C.	A Computer-Assisted Mathematics of Finance Course	88
Barrozo, T. Richards, H. Olsen, H.	Impact of CAI for Remediating Basic Skills Deficiencies of Open Admissions Students	91
Bork, Alfred	A Computer-Based Learning Aid in Electricity and Magnetism	97
Bork, Alfred Luehrmann, Arthur Schneider, Edward	Conference on Intelligent Videodisc Systems	100

Brown, Bobby R. Ellinger, Robert S.	The Whens and Hows of Computer Based Instructional Simulation	102
Burns, R. E. Davisson, W. I. Flanagan, C. M.	Computer-Managed Tutorials at Notre Dame.	109
Campanizzi, J. A.	Effects of Locus of Control and Provision of Overviews in A Computer-Assisted Instruction Sequence	114
Crane, Stanley A.	Eighteen Hands Are Better Than Two - The Multiplexer Approach to Computer-Assisted Instruction	119
Dennis, J. Richard	On the Construction and Use of Computerized Simulations	122
Fraser, Herbert W.	Using the Calculator Assisted Game Economy (CAGE): A Sixth Grade Experiment	126
Hallworth, H. J. Brebner, Ann	Computer Assisted Instruction and the Mentally Handicapped: Some Recent Developments	131
Hunter, Beverly Hargan, Carol	Student Accomplishments from Instructional Computing.	135
Huntington, John F.	Language Limitations for CAI.	144
Jones, Michael C.	TICCIT vs Lecture: An Evaluation of TICCIT Computer-Assisted English Composition and Mathematics Instruction.	149
Kirsch, Robert J.	Student Conducted Survey of the Opinions of Orangeburg Retail Business Managers, White and Non-White on Local Political Issues	161
Kline, Edward A.	Computer-Aided Review Lessons in English Grammar.	165
Lappan, Glenda Winter, M. J.	What Will Happen If __________?	170
Macero, Daniel J. McElroy, Francis C.	Low Cost CAI With A Microcomputer	175
Mason, George E. Blanchard, Jay S.	Computerizing Reading: A Review	181
Meyer, Rick	A Universal Dissemination for Computer Presented Courseware	187
Modesitt, Kenneth L.	Beyond Independence: A View of Games and Personal Computing	188
Parrish, Roy J. Jr. Beck, Henry J.	A Realistic Use of Computerized Methodology in Accounting Instruction .	194
Pollack, Richard A. Schwab, James W.	USNA-Computer Supported Instruction System: Sophisticated, Generative and Easy to Author.	199
Preskenis, Kenneth	Integrating Computer Use Into an Undergraduate Mathematics Course . . .	203
Prysby, Charles L.	Integrating Computer Based Survey Analysis Into Undergraduate Social Science Courses	206
Spencer, Larry T.	A Computer-Based Quizzing Program and its Use in Teaching Biological Subjects.	211
St. Andre, Richard Eggen, Maurice	Computer Projects in Undergraduate Calculus	215
Thompson, Patrick W. Cooney, Thomas J.	Constructing Computer Programs for Mathematics CAI Using A Pedagogical Model for Teaching Mathematics	220
Uhran, John J. Jr. Davisson, William I.	Modeling and Simulation as A Teaching Tool.	230

Wilde, Jane H. — Computer Assisted Instruction and History . . . 237

Yens, David P. — A Micro and Maxi Computer Network for Medical Data Base and Educational Activities . . . 242

COMPUTER-ASSISTED GUIDANCE

Kobylarz, Linda L. — The Guidance Information System: A Computer Based Career Education Tool . . . 248

Sampson, James P. Jr. — The Role of the Counselor in Relation to Computer-Assisted Career Guidance . . . 251

COMPUTER-ASSISTED TESTING

Chand, Donald R. — An Automated System for Grading Basic Programs . . . 253

Etersque, Stanley — Costs and Design Criteria for CATC Systems . . . 258

Hamilton, Gregory C. — Item Selection: A Computerized Decision-Making Process . . . 265

Keller, Lois J.; Williams, Robert L. — Test Generator . . . 270

McHenry, Hugh — Computer Assisted Testing Services . . . 274

Milner, Stuart D.; Wildberger, A. M. — The Evaluation of Computer Testing: Design Stategies and Feasibility . 279

Muiznieks, Viktors J. — Meaningful Test-Result Reporting . . . 286

Olympia, P. L. Jr. — Computer-Assisted Test Construction Under TITA2 . . . 293

Rubicam, Irvin; Olivier, William P. — A Non-Hierarchical Computerized Test Item Data Base . . . 297

Spero, Samuel W. — An Instructional Strategy for Computers and Calculators With Application . . . 302

Van Verth, James E. — A Versatile Basic Program for Test Generation Using A Small Time-Sharing Minicomputer System . . . 307

Warren, J. A.; Michener, R. D.; Bugbee, A. — Automating Exams for A Statistics Course: The Beginning of A Case Study . . . 310

Van Willis, W.; Seely, Oliver Jr. — Computer Assisted Test Construction: Two Years of User Experience With the SOCRATES System . . . 315

COMPUTER-DESIGNED INSTRUCTION

Hyde, Kendell — The Use of Computer-Assisted Lessons for Calculus . . . 318

Vickers, Frank D. — Total Computerization of A Course . . . 321

COMPUTER-MANAGED INSTRUCTION

Boas, Edward Jr.; Frantz, Nevin Jr.; Matthews, John — CMI for Staff Development in A Community College System . . . 327

Clock, Charles J. Jr.; Witt, Gail K. — A Generalized System for Classroom Instructional Management . . . 332

Conlan, Sharon — Design and Implementation of an Elementary Computer Interactive Reading Package and Spelling Package . . . 336

Folk, Mike — Computer Supported Precalculus: Solving Some Problems in Record Management and Test Generation 340

HARDWARE

Koide, George; Howard, R. Lee — A Microcomputer System for Behavioral Experiments. 345

SOFTWARE

Beidler, John; Meinke, John — Data Structures for Sophomores 348

Cashman, Thomas J. — Structured Design, Structured Programming, Structured Walkthroughs and the New Philosophy of Computer Programming. 355

GENERAL INTEREST

Allan, John F.; Stiglmeier, John — Procedures for Collecting Data From State and Local Education Agencies 361

Eddins, John M.; Floyd, Samual A. Jr. — Computer Assisted Research: A Bibliographic Data Base for Locating Early Musical Scores by Black Americans. 365

Geil, John — Braille and the Computer 368

Haugo, John E. — A Statewide Computing Consortium: Its Progress and Plans 375

Judd, Wallace — Factors Assuring an Evolutionary Role for the Computer in Curriculum . 378

King, D. Thomas — Metro-II: An Educational Computer Consortium Near Commencement 384

Meade, Roland L. — A Computer Simulation of Navigating an Aircraft Using VOR. 389

Klassen, Daniel L. — A Study of Computer Use and Literacy in Schools, A Project Progress Report 393

Pearson, R. R. — Computer-Based Instruction: Network Status in North Carolina 394

Roy, Thomas M. — Automation and the Teacher Education Program 397

Scherer, Donald R. — The 9900, 16-Bit Microcomputer in the Educational Enviornment. 402

Weinman, Howard — Computer Studies in the Junior High School: A Three-Year Curriculum Sequence. 407

LATE PAPERS

All papers submitted late, but in time for publication, appear beginning on 412

AEDS BOARD OF DIRECTORS

1977-78

PRESIDENT

Dr. Judith Edwards
Alaska Telecommunications
526 Main Street
Juneau, Alaska 99801
907/586-2722

PRESIDENT-ELECT

Mr. Brad Burris
Executive Editor-Testing
Scott Foresman & Company
1900 E. Lake Avenue
Glenview, Ill. 60025
312/729-3000

PAST PRESIDENT

Dr. E. Ronald Carruth
Asst. Dir. for Dist. Services
MN SD Data Processing Joint BD
1925 W. County RD B2
St. Paul, MN 55113
612/633-9110

SECRETARY

Ms. Jackie Reynolds (1980)
Asst. Dir., Computer Center
Atlanta Public Schools
218 Pryor Street, S.W.
Atlanta, Georgia 30303
404/659-4714

TREASURER

Dr. Delbert Repp (1979)
Director, EDP
Office Computer Services
2 Pleasant Avenue West
Lancaster, NY 14086
716/686-2038

* * * * * *

Mr. Edward Boorsma (1980)
Coordinator DP
Winnipeg SD #1
1577 Wall Street East
Winnipeg, Man., Canada
204/775-0231 x 215

Dr. John Hamblen (1979)
Chairman, Computer Science Dept.
University of Missouri - Rolla
Rolla, MO 65559
314/341-4492

Winston Addis
Supt. of School
Mt. Vernon Community Schools
Mt. Vernon, Iowa 52314
319/895-8845

Mr. Thomas Hartsig (ex officio)
Computer Science
Macomb Inter Sch Dist
44001 Garfield
Mt. Clemens, MI 48043
313/286-8800 x 171

AEDS BOARD OF DIRECTORS
(Continued)

Mr. Ralph Lee (1978)
Director, Computer Center
University of Missouri at Rolla
Rolla, MO 65401
314/341-4841

Sister Patricia Marshall (1980)
Director Computer Center
Xavier University
7325 Palmetto Street
New Orleans, LA 70125
504/486-7411 x 255

Mr. Paul Pair
Control Data Corporation
1650 W. Alameda
Tempe, AZ 85282
602/968-8791

Mr. Wm. R. TenEick (ex officio)
Dir., Academic Computer Svcs.
De Paul University
25 S. Jackson Blvd.
Chicago, Ill 60604
312/321-8336

Ms. Ann Tompkins (1978)
Mellonics Information Center
6701 Variel Avenue
Canoga Park, CA 91303
213/887-5100 x 257

EXECUTIVE SECRETARY

Ms. Shirley Easterwood
1201 16th Street N. W.
Washington, D. C. 20036
202/833-4100

* * * * *

16TH ANNUAL CONVENTION COMMITTEE

CONVENTION COORDINATORS

Thomas J. McConnell, Jr.
Asst. Sup. for Ed. Data Proc.
Baltimore City Public Schools
2330 St. Paul Street
Baltimore, MD 21218
301/396-6520

Jackie M. Reynolds
Assistant Director
Computer Center
Atlanta Public Schools
218 Pryor Street S. W.
Atlanta, Georgia 30303
404/659-4714

LOCAL ARRANGEMENTS

Mary Ann Stallybrass
Programmer, Computer Ctr.
Atlanta Public Schools
218 Pryor STreet, S. W.
Atlanta, Georgia 30303
404/659-4714

TREASURER

L. H. Tippett, Jr.
IBM, GSD
Suite 1100
100 Colony Square
1175 Peachtree Street, NE
Atlanta, Georgia 30361
404/885-6432

PUBLICITY

Aaron J. Johnson
Asst. to VP Fin. Affairs
Georgia State University
Atlanta, Georgia 30303
404/658-2320

PROGRAM

Chairperson

James E. Eisele
Office of Comp. Activities
University of Georgia
Athens, Georgia 30306
404/542-3106

Assistant Chairperson

John S. Edwards
Office of Comp. Activities
University of Georgia
Athens, Georgia 30602
404/542-3106

EVALUATION

Grover C. Simmons
Director
Computer Center
Atlanta University
P. O. Box 324
Atlanta, Georgia 30314
404/525-1501

EXHIBITS

James Hauther
Technical Support Supervisor
Department of Information Systems
DeKalb County School System
3770 North Decatur Road
Building A
Decatur, Georgia 30032
404/296-2000 x 234

EDITORIAL REVIEW COMMITTEE

Richard Pogue, Chairperson
Stanley Etersque
Robert Hashway
Alan Jones
David Kneifle
Lu Penn
Pier Paolo Pisani
Dennis Sorge
John Wilcox

COMPUTER TECHNOLOGY: THE EDUCATIONAL CATALYST

NOTE

The papers which are printed in this volume represent a significant contribution from those who participated in the 16th Annual AEDS Convention held from May 15-19, 1978, in Atlanta, Georgia. Unfortunately, a few papers which were presented at the Convention could not be printed in this collection for a variety of reasons. The Program Committee is equally grateful to those who delivered papers which do not appear here. If the reader is aware of such a presentation and is interested in further information about that paper, it is recommended that the presenter be contacted directly.

Additional copies available for
$10.00, prepaid, from:

Association for Educational Data Systems
1201 Sixteenth Street, N.W.
Washington, D. C. 20036

Library of Congress Catalog Card Number ISSN 0147-9296

Copyright © 1975, the Association for Educational Data Systems

ADMINISTRATIVE APPLICATIONS

EDUCATIONAL COMPUTER CENTER SECURITY

Michael Alverson
Peat, Marwick, Mitchell & Co.

The development and application of a data processing security evaluation program provides the basis for planning, reviewing and improving data processing security and for dealing with the problems of computer security faced by today's educational computer centers.

INTRODUCTION

Data processing security has become a matter of growing concern. Reflecting the exponential increase in the use of computers in our modern life style, educational institutions have made substantial resource commitments for data processing hardware, software, people and services. Almost every major educational institution has one or more computer centers. The pervasive influence of data processing and computers has made administrators conscious of the need to effectively use their computer resources and protect against their misuse.

The problem of computer security has been heightened by rapid technological advances and the increased complexity of computer centers, as evidenced by the use of large scale computer hardware, communications networks, complex operating systems and application systems designed with a high degree of sophistication oftentimes using data base management techniques. The increased ability of computers to furnish a wide range of processing capabilities has brought on a high degree of centralization in both data and processing facilities. The complexity and interdependence in an educational computer environment opens many opportunities for security threats. The result can be purposeful (and unnoticed) misuse of the data processing environment, neglect in the area of controls, audit trails or security, resulting in the unnecessary exposure of data processing resources to avoidable risks.

EXTENT OF THE PROBLEM

Security of an educational computer center should be viewed from a total system perspective in order to appreciate the magnitude of the problem. Three aspects of this security problem are especially significant:

- No university can function effectively as a business entity without the availability of computer services on a continuing basis.
- The value of the resources employed by university computer centers, and their associated equipment and systems are significant assets that should be properly safeguarded.
- The amounts of potentially sensitive information stored in computer and data bases should be classified and provided appropriate security treatment.

The unquestionably large amounts of money associated with educational computer centers justify a thorough analysis of data processing performance and operations to assess whether unnecessary risks are present that could seriously degrade the University's ability to carry out its mission. Even relatively minor improvements in the operating efficiency of such a "high value" activity can yield substantial benefits in increased performance or savings.

Heightened awareness of confidentiality and privacy issues has been focused on data processing procedures and data bases. Today, the educational computer center must respond with effective programs that appropriately protect sensitive data to a greater degree than required previously. New legislation can have an immediate impact if not planned for.

Thus, an educational computer center faces a situation in which considerable resources and sensitive data are or can be exposed to the risks of damage, misuse, improper control or poor management. That these risks are real and substantial is made clear with every new revelation of computer security failures that result in substantial losses for the affected organization.

SEARCH FOR A SOLUTION

Having identified the problem of data processing security and having discussed the scope of risks it entails, it would be useful to discuss an approach to its solution. Unfortunately, solution is too final a term and something that is not really attainable. For, like Diogenes searching for an honest man, one could search forever for a completely secure computer center. However, actions can be taken to reduce the various risks from improper data processing security. To proceed in this manner, a comprehensive data processing security evaluation program should be developed.

The purpose of such a program is to determine the extent and adequacy of computer security. This determination can then be used as the basis for an action plan designed to improve the data processing security and overall performance of the educational computer center. A data processing security evaluation program should contain certain desirable features, including the following:

- comprehensive and in-depth evaluation of applications, communication networks, terminals, information flow, plant and facilities, administration procedures and other security concerns;
- quantified description of the security of a given data processing installation, based on an embedded profile and scoring procedure, in order to consider the impact of change in a given data processing environment from year to year; and
- sensitivity to the multidimensional character of educational computer centers striving to meet the requirements of various constituencies, e.g., administration, and education.

Comprehensive Evaluation

The first desirable feature, comprehensive and in-depth evaluations, reflects the total system perspective that is required for an effective evaluation program. Just as the problem of data processing security should be viewed in total, so too should the response. Like a chain that is only as strong as its weakest link, the security of an educational computer center is only as strong as each of the elements of that security. For any computer environment to be considered strong in the area of security, it must be strong in each of the following elements:

- Internal Audit
- Administrative Security
- Physical Security
- Standards
- Processing Security
- Operating System Security
- Software Security
- Data Base Security
- Communications Security
- Application Security

Each of the interrelated elements of computer center security need to be evaluated in order to present a truly comprehensive and in-depth evaluation.

Quantification

The second desirable feature of the data processing security evaluation program, that it should be quantitative, is an outgrowth of the idea that data processing security can be measured along a continum from more to less secure computer center environments. If an evaluation is performed using an embedded profile and scoring procedure, a benchmark is established for a given computer center at a particular point in time. This benchmark provides the basis for monitoring improvements or heightened risks in overall security of a computer center.

The embedded profile can be constructed with the ten previously mentioned elements as a starting point. Within each element multiple levels of detailed questions can be constructed that characterize the security effectiveness of that element. Assignment of numeric values from a limited range for each specific question response provides some flexibility in the scoring procedure. As in any evaluation scheme, the weightings assigned to each element are critical. These weightings combined with the numeric values provide a total score. For that score to be most meaningful, the weightings assigned should reflect the perceived likelihood of the various security threats and the value of the assets at risk.

For example, an educational computer center might want to assign heavier weightings to Physical Security to emphasize that particular aspect of the data processing security environment.

Educational Environment

The third desirable feature of the data processing security evaluation program, sensitivity to the multidimensional nature of educational computer centers, is an extension of the first two features of comprehensive evaluation and quantified scoring. In order to most effectively evaluate the data processing security of an educational computer center, the methodology used must reflect the special circumstances and multiple constituencies that such a center must serve.

Some progress toward this goal can be achieved by paying special attention to the evaluation, e.g., inventorying all users of the center and evaluation of all systems. In this way, no groups significant to the data processing security evaluation are omitted.

In addition, the weighting assigned to various elements should reflect a sense of balance between conflicting data processing security objectives of the various constituencies of the educational computer center. This weighting process should be a participative exercise where administrative, data processing, campus security officers, department heads and other knowledgable members of management are involved in the weighting process and reach agreement as to which areas present the greatest risk.

IMPLEMENTATION ISSUES

The decision to develop a data processing security evaluation program is an important one with potentially widespread impact on the educational computer center. Associated with the implementation of such a program are several issues, including the following:

- resources committed - the preparation for and participation in such an evaluation program requires substantial professional time and skills to complete the depth of the survey required. This expenditure of effort should be weighed against the exposure to risk represented by an unexamined data processing environment.
- future systems development - the increased awareness of data processing security considerations should be formally incorporated into the computer center's system development methodology. A suggested technique is the development of a security implications report in parallel with initial design efforts;
- periodic evaluations - once the data processing security evaluation program has been established, and scored on an individual area and overall program basis, additional consideration can be given to follow-up security reviews, perhaps on a reduced-effort basis, to survey the security impact of new applications becoming operational, changes in equipment configurations, and administrative or organizational changes.

16

A COMPUTER SYSTEM FOR FINANCIAL AIDS

BY

CAROL AVELSGAARD
SENIOR PROGRAMMER
BEMIDJI STATE UNIVERSITY
BEMIDJI, MN. 56601

The major objective of the Financial Aids Program at Bemidji State University is to provide funds to all students who would not otherwise be able to attend college. In doing so, all possible sources of funds and student employment have been placed under the central coordination of the Financial Aids Office. It was felt that in order to handle the high volume of applicants for financial aid and to allow the counselors to spend most of their time counseling students rather than counting them, a computer system was necessary to do as much of the detailed record keeping, need analysis, and routine packaging as possible. The system was designed to be able to meet the requirements of Federal, State, and Institutional accounting, reporting, and applications for funds.

A. Collection of Data Elements

Students who apply for financial aid at Bemidji State University complete the ACT (American College Testing) financial statement and a brief BSU application form containing demographic information and student employment preferences. The ACT form also serves as application for the Minnesota State Scholarship and Grant Program and for the Basic Educational Opportunity Grant. Student financial data, family financial data, and the results of ACT need analysis are returned to the Bemidji State University Financial Aids Office on standard CFARS for any student who has indicated an interest in attending BSU.

Punched cards are also received from ACT and loaded into the BSU Financial Aids System. The student's application enters a tracking process, which screens it for missing data and prepare letters to the student indicating the status of his application. When the application is complete, a combined counseling report and data entry form is printed which includes both current and historical values of financial data, need analysis results, packaging results, and employment asignments. (See Figures A-C) During the academic year demographic elements are transferred quarterly from the Student Records System to the Financial Aids System.

B. Evaluation.

The document received by the counselor is designed to help identify changes in the student's financial situation from year to year and to act as a coding form for changes, corrections, and evaluation results. (See Figure B) The BSU Financial Aids System includes a routine that performs ACT need analysis. If a correction or change is made to a need analysis input item, after receiving a tax form or notification of a mid-year change in a student's family income situation, the counselor can request that the need analysis routine be repeated to reflect the new values.

A preliminary award package is prepared by the system. The counselor has the ability to change any dollar amount or quarterly distribution. Decisions on allocations of funds are made with the aid of a fund summary which is printed at the end of each update cycle, showing budgeted and allocated award totals by fund and employment type.

C. Packaging Routine.

Funds packaged initially by the system are Basic Educational Opportunity Grants (BEOG), Minnesota State Scholarships and Grants, Minnesota State Indian Scholarships, Tribal Scholarships, Institutional and Foundation Scholarships, Supplemental Educational Opportunity Grants, National Direct Student Loans, Guaranteed Student Loans, and student employment, which includes Minnesota Workstudy, Federal Workstudy, and institutional operating accounts. The routine examines student eligibility for these funds and allocates the appropriate award amounts until the student's need is met. If a fund is not available, the Financial Aid Office is able to instruct the packaging routine to bypass it.

Counselors may modify the results of the initial packaging or request that the system generate a new package. (See Figure C) The system is designed to take care of the routine allocations of aid while allowing the counselors complete flexibility in tailoring a package to fit an individual student's needs.

D. Award Letters

Computer generated letters are sent to students informing them of their approved aid package and of any subsequent changes. Preliminary award notices are provided for new entering students through Spring Quarter and final award letters are generated after Minnesota State Scholarships and Grants have been awarded by the state. tudents who have not been accepted for financial aid are sent a typewritten letter to avoid the I-HAVE-BEEN-REKECTED-BY-A-COMPUTER syndrone. A National Direct Student Loan signature form and a Basic Educational Opportunity Grant certification form are generated for students receiving these funds to sign and return.

The letters include a standard description of the award package with free-form explanatory paragraphs which can be modified by the Financial Aids Office to fit up to sixty different award categories or processing steps. For rejected applications the explanation is a note of instruction to the typist. Current student addresses are obtained from the Student Record System.

E. Check Writing and Reconciliation

Prior to the start of each quarter, award checks are prepared for all students who have approved awards for that quarter. The checks are printed by the Check Writing System, which also makes entries to the University Accounting System and the Check Reconciliation System. After pre-registration for the quarter is complete, the checks are distributed to students displaying a valid fee statement. Each student normally receives one check for the total amount of his award for that quarter. Financial aids checks are also produced daily throughout the quarter to cover new awards and changes in a student's income or expenses.

The Check Writing System produces a card containing the student's social security number, the fund number, and the check number for each separate fund included in the total award. This card re-enters the Financial Aids System as a record of funds received by the student. It also enters the Loan Repayment System if it represents a student loan. Each quarter a report is printed comparing awards allocated to awards received, which allows the Financial Aids Office to monitor the disbursement of funds throughout the year.

F. Retrieval

The basic management report used by the Financial Aids Office is a one-line summary by student which includes a few demographic items, the status of the award, and the the breakdown of the award total into categories. This is produced weekly and is used to answer the most common day-to-day inquiries by students and by other administrative offices. Other report formats produced by the retrieval system include: a detailed list of students' allocated dollar amounts by fund type, a detailed list of students employment records including hours worked and dollars earned, a summary of dollars and numbers of students, mailing labels, call cards for use by other programs, and a list of students whose need has been exceeded. Any of these standard report forms can be obtained for virtually any subset of students on the file.

There are no programs in the system which are written specifically to produce fiscal operation reports or applications for funds. One program edits Federal funds to identify students with incomplete information. Data for annual reports is then gathered from several different reports produced by the retrieval system. This replaces the process of re-programming to satisfy new requirements by re-running a program which is known to give accurate results. This, together with continual monitoring of the system through daily summaries and monthly check reconciliation, has simplified the task of annual reporting.

G. Student Employment

Students eligible for part-time employment are assigned to departments within the university by fund type - Federal Workstudy, Minnesota State Workstudy, or institutional operating accounts. Time cards are produced by the system every two weeks, distributed to the departments, filled out by the students, and re-loaded after the total hours worked have been keypunched. Checks are produced by the Student Payroll System, which also provides the necessary entries to the Accounting System. Each department receives a summary report by student, showing hours worked and hours assigned, with the time cards. The Financial Aids Office receives a detailed report by student within department, showing hours worked, hours remaining, dollars spent, and indicating whether a student is working more or less than the anticipated number of cumulative hours for the quarter. This report is also summarized by department, so that the campus-wide distribution of student workers and fund utilization can be monitored.

The production of time cards is controlled by a starting date and an ending date in the student's employment record. These dates are ordinarily assigned through a master control card by the system, but they can be altered for any individual student. A part of the turn-around document produced when the students' application is complete includes a report of his entire student employment history. (See Figure A) This is used by the counselor to determine the starting salary. The system includes a means of adjusting the hourly wage of all students when changes are made to the minimum wage.

H. On-Line Inquiry

All updating of the file is done through the batch system. However, the Financial Aids Office has on-line display capabilities. Financial Aids records, Student Record System files, Accounting System entires, Student Payroll System files, and Accounts Receivable System records can all be accessed in this manner.

The Financial Aids System uses one indexed sequential mass-storage file, which includes the entire history of each active student. All reports are keyed by fiscal year, since the Financial Aids Office is normally processing two different fiscal years at any given time. (They are, for example, producing checks for one year while either assigning aid for the next year or preparing summary reports for the previous year.) When all the reporting has been completed for any fiscal year, the file is purged, and the records for any student who has not been active for that year or the two previous years is removed from the active file. All records are retained on a history tape. So far, the active file has remained manageable in size.

The file contains, in addition to student records, control records for funds and fiscal years which include such things as the title of a fund, total dollars available in a fund, universal starting dates for student employment, control flags for the packaging routine, and control flags for the check writing program.

Statistics

Computer	UNIVAC 1106
No. of students on file	10,812
Active students in fiscal year 1978 . . .	4,297
No. of records on the file	91,218
Total dollars	4.6 million
No. of mass storage tracks (10,752 chtrs. per track)	2,449
No. of programs	45
Maximum core size req'd. (36 bit word) . .	44K
Financial Aids Staff (Awarding)	
Director	1
Counselors	3
Clerical Staff	4
Student Workers	3
Computer center staff assigned to Financial Aids	1/4
Average SUP (Standard Unit of Processing) time per month	22 hours
Average keypunch time per month	22 hours

FIGURE A

```
   NEED ANALYSIS

BEMIDJI STATE UNIVERSITY                                                         FISCAL YEAR 78
FINANCIAL AIDS                                                                   AS OF 03/10/78
EMPLOYMENT RECORD - 25003

  TOT    TOT    PY CUMULATIVE    F   P          EST    START       END
  HRS    AMT    PD AMOUNT   PD   YR  D   DPT    AMT    DATE        DATE      PROC TP  HRS  WKS RATE
                                     *   ***   17-1    22-27       28-33     34-5 36 37-8 39-0 41-3
220.0  445.41 20  445.41 20 E.75. 1 .204. 588. 09/03/74. 05/23/75.      . W. 10 . 31 .2.00.
324.0  694.80 19  694.80 19 E.76. 1 .154. 660. 09/02/75. 05/21/76.      . W. 10 . 33 .2.20.
338.0  814.80 20  814.80 39 E.77. 1 .154. 792. 09/07/76.   /  /  .      . W. 10 . 33 .2.40.
 40.0  116.00  1  930.80 40 E.77. 2 .154. 100. 11/22/76. 11/29/76.      . W. 40 .  1 .2.90.
 53.0  133.75  2 1064.55 42 E.77. 4 .154. 150. 06/06/77. 06/30/77.      . W. 15 .  5 .2.50.

221.0  582.25 13  582.25 55 E.78. 1 .154. 550. 09/03/77.   /  /  .      . W. 10 . 22 .2.85.

 39.0  103.49  1  685.74 56 E.78. 2 .154. 100. 11/21/77. 11/28/77.      . W. 40 .  1 .2.62.

                  685.74 56 E.78. 3 .154. 100. 03/06/78. 03/10/78.      . W. 40 .  1 .2.89.

 84.0  211.25  4  896.99 60 E.78. 5 .154. 150. 07/01/77. 08/12/77.      . W. 15 .  5 .2.85.

                            E.    .     .    .    /  / .    /  /  .      .  .    .    .    .
```

NEED ANALYSIS

FIGURE B

```
BEMIDJI STATE UNIVERSITY
FINANCIAL AIOS                                                                          FISCAL YEAR 78
FINANCIAL STATEMENT - 25003                                                             AS OF 03/10/78
                                              74-75   75-76   76-77   77-78 B 78  REVISED NA PK
STUDENT INCOME:   EST EXPENSE                  2400    2792    5273    6057   01 .  15 - 20 21 22
                  EST GROSS INCOME                      305            5706   02 .        .  .  .
A SEX 2           FEDERAL TAX                                           239   03 .        .  .  .
  RACE 5          STATE TAX                                             192   04 .        .  .  .
  HSR 73          HON INST WAGES                                        960   05 .        .  .  .
  TEST 025        SPOUSE WAGES                                 1800           06 .        .  .  .
  ADM             SUMMER SAVINGS                100             400     300   07 .        .  .  .
  GPA 3.27        TAX REFUND                                                  08 .        .  .  .
  HRS 191         GI BILL                                                     09 .        .  .  .
                  OTHER VA BENEFIT                                            10 .        .  .  .
D NW/TR 2         VOC REHAB                                                   11 .        .  .  .
  DGR 3           SOC SEC                                                     12 .        .  .  .
  RES 2           OTHER INCOME                  762                           13 .        .  .  .
  MAR 2           CASH                          330                    2000   14 .        .  .  .
  CHLD 0          HOME VALUE                                                  15 .        .  .  .
  RANK 4          HOME LIABILITY                                              16 .        .  .  .
  INDP 1          F/B VALUE                                                   17 .        .  .  .
  AGE 20          F/B LIABILITY                                               18 .        .  .  .
                  MOTOR VALUE                                           800   19 .        .  .  .
                  MOTOR LIABILITY                                             20 .        .  .  .
                  OTHER ASSET                                                 21 .        .  .  .
                  OTHER LIABILITY                                             22 .        .  .  .
                  OTHER DEBT                             14                   23 .        .  .  .

FAMILY INCOME:    ADJ GROSS INCOME             1769    1627           12430 C 78
                  HOTHER INCOME                                               01 .        .  .  .
C PAR AGE 00      NON TAX INCOME               3232    3936                   02 .        .  .  .
  PAR NO 2        FEDERAL TAX                                          1235   03 .        .  .  .
  DEP  0          OTHER TAX                            1332                   04 .        .  .  .
  DEP IN HE 1     MED-DENT EXPENSE              954    1700             586   05 .        .  .  .
                  OTHER DEDUCTIONS             1100                           06 .        .  .  .
                  MISC EXPENSES                2442      64            2069   07 .        .  .  .
                  F/B DEPRECIATION             4945    5247                   08 .        .  .  .
                  FARM ACRES                    385     400                   09 .        .  .  .
                  HOME VALUE                                          27200   10 .        .  .  .
                  HOME LIABILITY                                       1948   11 .        .  .  .
                  F/B VALUE                   52200   52200                   12 .        .  .  .
                  F/B LIABILITY               28000   26000                   13 .        .  .  .
                  MOTOR VALUE                   500     100                   14 .        .  .  .
                  MOTOR LIABILITY                                             15 .        .  .  .
                  OTHER ASSET                 14500    1100             139   16 .        .  .  .
                  OTHER LIABILITY             11200    3900                   17 .        .  .  .
                  PARENT CONTR                                                18 .        .  .  .
                                                                              19 .        .  .  .

NEED ANALYSIS:    BSU STDNT CONTR               945      95    2200    1626 D 78
                  ACT STDNT CONTR               945     345                   01 .        .  .  .
                  CFAR STDNT CONTR             1145            3150-   2170   02 .        .  .  .
                                                                              03 .        .  .  .
                  BSU PARENT CONTR                                            04 .        .  .  .
                  ACT PARENT CONTR                                            05 .        .  .  .
                  CFAR PARNT CONTR             437-                           06 .        .  .  .

                  BSU NEED                     1455    2792    3073    4431   07 .        .  .  .
                  ACT NEED                     1455    2447    4973    4914   08 .        .  .  .
                  CFAR NEED                    1720            3900    3453   09 .        .  .  .
```

FIGURE C

```
    NEED ANALYSIS

BEMIDJI STATE UNIVERSITY                                                           FISCAL YEAR 78
FINANCIAL AIDS                                                                     AS OF 03/10/78
AWARD DISTRIBUTION - 25003

INSTITUTIONAL:    CUM 75-6 76-7 77-8 G 78  AWARD CHECK LT  SSII  FALL  WTR   SPR   SSI
                                           15-19 20-24 25 26-30 31-35 36-40 41-45 46-50

NDSL      1                     850   10 .     .     . .  175.     .     .  300.  375.

SEOG      2       350           750   15 .     .     . .     .  215.  214.  214.  107.

BSU SCH                               20 .     .     . .     .     .     .     .     .

OUTSIDE AID :

BEOG              850 1062 1062       40 .     .     . .     .     .     .     .     .

MN SCHOL                              50 .     .     . .     .     .     .     .     .

MN GRANT          850 1050  725  675  52 .     .     . .     .  225.  225.  225.     .

MN IND GRT                            60 .     .     . .     .     .     .     .     .

TRIBAL GRT                            61 .     .     . .     .     .     .     .     .

FISL                                  70 .     .     . .     .     .     .     .     .

OTH SCHOL                             80 .     .     . .     .     .     .     .     .

TOTAL AID:

GIFT AID          2050 2112 1787 1425                             440   439   439   107
EMPLOYMENT         445  694 1041  864
LOAN                              850                       175               300   375

TOTAL             2495 2806 2828 3139                       175   440   439   739   482
UNMET NEED                        1292

                                    D 78  DEPT  RATE  HRS   WKS   TYPE  PRD
                                          15-17 18-20 21-22 23-24  25    26
   NA        PROC      LETTER  REJ  00  10 .     .     .     .     .     .     .
  DATE       DATE       DATE   PROC 15        SSII  FALL  WTR   SPR   SSI
04/01/77  02/22/78  02/23/78   CANC 00         15    16    17    18    19
                               AUTO  1  11 .     .     .     .     .     .
                               APPR           REJ   PROC  CANC  AUTO  APPR   DATE   COUNS
                               COUNS J        15-16 17-18 19-20  21    22    23-28   29
                                        12 .     .     .     .     .     .  /  /  .     .
```

THE APPLICATION OF MIS CONCEPTS TO COURSE PLANNING

Stanley J. Birkin
University of South Florida

The problem of planning course offerings at colleges and universities has long been recognized as a difficult one. Often decisions concerning detailed schedules must be made long before there is any adequate available data on budgets, resources (classrooms, instructors) and most important the needs of students. It is quite common for administrators to be required to plan ahead for one or two full academic years and often such temporary and tentative plans can easily become firm commitments.

The purpose of this paper is to report on the approaches which have been used to tackle this problem in the College of Business at the University of South Florida. The system which has evolved uses a four quarter "cradle-to-grave" data base approach which keeps track of a given schedule from its initiation to its final disposition.

INTRODUCTION

The creation of future academic course schedules presents a challenging task for many administrators. Flexibility in curricula requirements and the uncertainty of students' course needs result in a situation where the questions as to the types of courses and the number of sections to offer often must be made with only historical data to use as a basis. Only in a few instances have institutions begun to "block schedule" course offerings, a system which reduces the uncertainty of students' course needs. For the majority of cases, however, an administrator repeatedly faces the scheduling problem and must plan ahead by "looking backwards" for guidance. This paper reports on a series of experiments which have been undertaken at the University of South Florida to tackle this problem using a system called "PLANNER".

THE MODEL

At the heart of the "PLANNER" system is a four-quarter data base of future course offerings. Essentially, this data base contains data on each course -- and each section of every course -- that will ever be offered in the future. For each course offering there is a series of codes indicating which of the future quarters that course will be offered. For example, a Thursday night section of a given course may be indicated as being offered in Quarters II and IV, whereas a day section of an upper-level course may be indicated as only being offered once per year in Quarter III. The entry for a given offering also includes other data such as class size, format, times, and possibly the location of the class and the instructors. This four-quarter data-base is constantly updated to reflect permanent changes in curricula and/or course requirements. Changes which are short-run in nature -- such as a course cancelled because a given faculty member is on leave-of-absence -- are not posted to the data base. Another source of updated information comes from past schedules. As actual enrollment in various sections is known and class caps are changed, these are evaluated and posted to the data base, if the change is considered to be representative of a shift in student enrollment in that section.

When the time comes for the initiation of a future quarter's schedule, the PLANNER system retrieves the planned course offerings as indicated by the current status of the four-quarter data-base, and generates an "on-line" free of these course offerings. This "planned" future quarter's schedule is then scrutinized by the appropriate chairperson or administrator. Final changes to this file are posted to this "planned course offerings" file to accomodate last minute changes in available resources (faculty, classrooms, etc.). The resulting file is used to initiate the offerings for the official university master schedule for that quarter.

OPERATION OF THE PLANNER SYSTEM

The updating of the course files in the PLANNER system is performed using an "on-line", interactive system called UPDATE. This is a "user-oriented" system designed to be operated by clerical staff with no computer background. The system can be operated using a CRT or hard-copy terminal. The UPDATE system is invoked for a given future quarter and allows the clerical operator to post changes using "change codes". These allow for the changing of each data item in the record, for example, by typing in: 18/0164.

The operator is requesting a change of instructor (code 18) for course reference number 0164. The UPDATE system will then respond by requesting the name of the new instructor for that course. Alternatively, for the novice clerical operator, it is possible simply to request a change code 18 and let the system prompt first for the course reference number, and then for the instructor name. Other change codes allow for the deletion of a course that is cancelled, and for the inclusion of new courses that are added after the creation of the initial schedule. Following the initial university's registration process, this file is then updated with the latest enrollment figures. This usually occurs 4-6 weeks before the actual beginning of that quarter.

In this form, this file reflects the most accurate status of a future quarter's schedule. As such it is used for a variety of management reports and working documents. Some of the important ones are:

1. Course Offerings Report

This report provides for faculty, students and administrators, the current status of each offering. This can be used by administrators to decide on last-minute course additions or deletions or faculty reassignments. Also, a copy is posted by the Advising Office and is used by students needing to add courses late since it indicates those courses with available seats that are "open".

2. The "Tally-Sheets"

This report is used during the "drop-add" period at the beginning of each quarter. The "tally sheets" provide for each course a mark for each vacant seat in the class. As late-add students are enrolled the marks are crossed out to keep track of vacant seats. In a similar way accounting is made for students "dropping" courses during this time period.

3. Graduate Advising Sheets

For the graduate Advising Office, a set of advising sheets is prepared and used to plan courses to be taken in future quarters.

4. Off Campus Courses

Since the University offers courses on three branch campuses as well as the main campus, this report lists offerings on these campuses separate from the main campus.

5. Evening Course Offerings

The University has a substantial enrollment in classes during the evening and this report lists only these courses.

6. Instant Productivity Reports

At any time, the PLANNER system can produce college productivity reports for the various departments and levels, and by faculty member.

PROBLEMS, POTENTIALS AND THE FUTURE

The PLANNER system, which started out on an experimental basis three years ago is now in constant operation. Because the system is managed internally, it is more responsive to the user's needs. Also in contrast to the official university system it has the ability to provide timely reports for management decisions. The few problems that have been encountered have been personnel problems in training new clerical operators. In spite of its "user orientation", personnel who are totally unfamiliar with computers are often apprehensive working for the first time with such a system. Problems such as system failures in the middle of an UPDATE can cause concern, even though the PLANNER system provides total, automatic backup for such circumstances. For the future, the PLANNER system offers the opportunity for development of additional reports and analysts. Much development in this direction is underway and will be reported in future papers.

DATA FLOW IN A STATEWIDE HIGHER EDUCATION MANAGEMENT SYSTEM
THE NEED FOR A TWO-WAY STREET

Charles H. Bryson
and
Glynton Smith
Georgia State University

This paper describes a statewide data collection system established by a board of controls for postsecondary institutions and explains how this system relates to one institution, addressing these specific points: 1) Statewide basic data flow concepts; 2) Five sub-systems of a consolidated reporting approach designed to present a dynamic view of activities and resources; 3) The data flow from organizational areas at one institution to each of the state-level sub-systems; 4) The data clearing-house function; and 5) Perceptions on how improvements need to be accomplished.

The Carnegie Commission has described institutions of higher education as the most complex organization known to man (Perkins, 1973). Data collection systems interacting within these complex organizational structures are difficult to conceptualize and operationalize into practical and workable management tools. Herein lies a dilemma which many states are addressing, mindful of the fact that resources are limited and needs must be weighed carefully in this economic era.

The purpose of this paper is to describe a statewide data collection system established by a board of controls for postsecondary institutions and also to explain how this system relates to one institution. The paper addresses these specific points:

1. Statewide basic data flow concept.
2. Five sub-systems of a consolidated reporting approach designed to present a dynamic view of activities and resources.
3. The data clearing-house function which interfaces with both institutional and state-level personnel.
4. The data flow from organizational areas at one institution to each of the state-level sub-systems.
5. Perceptions on how improvements need to be accomplished to develop the data collection system aimed at (a)communications linkage and humanistics, (b) data collection practicality and reliability, and (c) assistance to decision makers at multilevels of management.

Statewide Data Flow Concept

State boards of higher education typically have viewed a statewide management reporting system as the vehicle for achieving greater efficiency and establishing improved accountability. These statewide data-gathering systems have taken approximately as many forms as the number of states within the United States (Laughlin, 1975). The complexities of higher education are so enormous that to define, measure, and explain reality represent attempts laden with limitations frequently inconceivable to or misunderstood by institutional administrators, faculties, governing boards, legislative bodies, and the public. It seems that the goals and objectives of higher education have not developed to the stage that all outspoken persons agree upon what questions need to be asked and answered to evaluate the performance of individual institutions or groups of institutions. These differing viewpoints converge upon a management reporting system and have significant underlying influence in the directions of data collection, analyses, and explanations. Many statewide systems have approached the problem by having institutions assemble data with the assumption that the availability of answers will somehow encourage the development of the potentially "right" questions.

About five years ago the Georgia higher education board decided it would be worthwhile to undertake the development of a comprehensive management reporting system which would strengthen its position with the legislative appropriations committee as well as enhance its ability to guide effectively public higher education in Georgia. Whereas, each of the thirty-two institutions, ranging from small junior colleges to large complex universities, had reported its activity in its own autonomous way; the management information system imposed some degree of consistency and structure to the wide variety of information flowing into the central governing board. A description of this management information system follows.

The Consolidated Reporting System (CRS)

The Consolidated Reporting System (CRS) is comprised of an interdependent set of five reports which present a dynamic view of each institution's academic activities, personnel, and physical resources. Chart 1 provides the basic flow concept of this reporting system.

The reports submitted to the CRS are as follows:

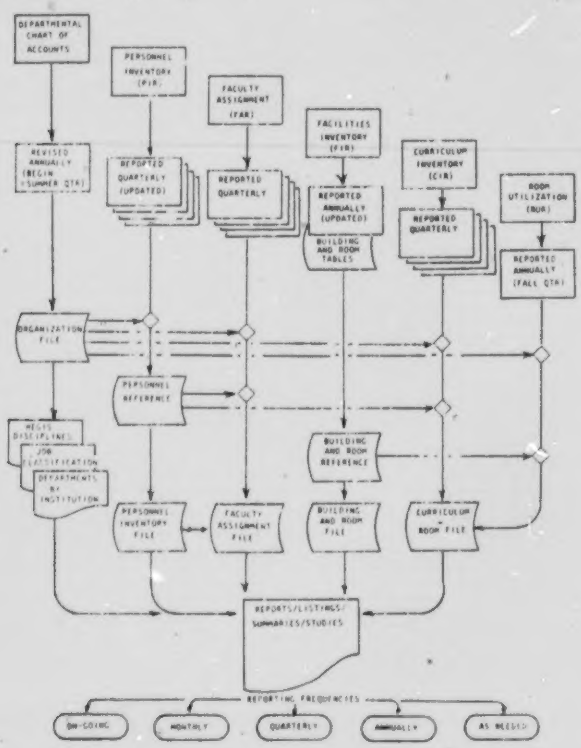
1. Personnel Inventory Report (PIR). This report, maintained on a continuing basis, provides the state board with an inventory of the personnel employed by the institution. For each individual employed in selected job classifications (job titles), the system reports basic personal information, budget department and position, title, and source of funding. The PIR information provides the state board with a continuing and detailed picture of the university's personnel resources.
2. Faculty Assignment Report (FAR). Encompassing personnel in many job titles beyond the academic faculty, this report basically indicates what the individual's job assignments are by category and time (instruction, research, public service, and administration). The report, completed and certified by the immediate supervisor on a quarterly basis, details the individual assignment of personnel resources by task category.
3. Curriculum Inventory Report (CIR). Submitted on a quarterly basis, this report presents a profile of every class section for which students are registered for academic credit.
4. Facilities Inventory Report (FIR). In this report the physical space available for use within the university is described in minute detail as to characteristics, facilities, and spatial dimension.
5. Room Utilization Report (RUR). For each class section in which students are registered, this report indicates the rooms and times scheduled.

This comprehensive set of five reports as outlined above provides an abundance of summary and detail information

CHART 1

BASIC FLOW CONCEPT

CONSOLIDATED REPORTING SYSTEM


DEPARTMENTAL CHART OF ACCOUNTS
PERSONNEL INVENTORY (PIR)
FACULTY ASSIGNMENT (FAR)
FACILITIES INVENTORY (FIR)
CURRICULUM INVENTORY (CIR)
ROOM UTILIZATION (RUR)
REVISED ANNUALLY (BEGIN SUMMER QTR)
REPORTED QUARTERLY (UPDATED)
REPORTED QUARTERLY
REPORTED ANNUALLY (UPDATED)
BUILDING AND ROOM TABLES
REPORTED QUARTERLY
REPORTED ANNUALLY (FALL QTR)
ORGANIZATION FILE
PERSONNEL REFERENCE
HEGIS DISCIPLINES
JOB CLASSIFICATION
DEPARTMENTS BY INSTITUTION
BUILDING AND ROOM REFERENCE
PERSONNEL INVENTORY FILE
FACULTY ASSIGNMENT FILE
BUILDING AND ROOM FILE
CURRICULUM - ROOM FILE
REPORTS/LISTINGS/ SUMMARIES/STUDIES
REPORTING FREQUENCIES
ON-GOING
MONTHLY
QUARTERLY
ANNUALLY
AS NEEDED

to the board of control regarding the operational activity of the institution. Analysis of university data, along with that from the other institutions in the Georgia system, not only provides summary information at the state level, but also allows detailed comparisons between institutions of academic schools, departments, and individuals. How well personnel resources are distributed and utilized can readily be ascertained through matching the detailed data of the PIR, the FAR, and the CIR. Similar analysis between the CIR, the RUR, and the FIR reveals the efficiency with which physical resources are allocated and used. The reports enable teaching workloads to be examined and compared at levels of institution, academic school, department, discipline within department, rank, and by individual.

Standard quarterly reports generated by the CRS provide the following summary information and institutional comparisons for use by the board's staff:

- Number of Different Courses Taught
- Number of Different Sections
- Student Credit Hours Produced
- Average Size of Sections
- Percent of Student Credit Hours by Taxonomy
- Percentage Distribution of Sections According to Section Size
- Percent of Student Credit Hours by Level
- Percent of Total Student Credit Hours Taught by Faculty Rank
- Percentage Distribution of Faculty Time by Academic Elements
- Percentage Distribution of FTE Faculty by Academic Elements
- Number of Faculty Members Reported
- Percentage Distribution of Faculty Members Reported by Institution
- Section Starting Time Report
- HEGIS/Department Classification
- Space Utilization Survey Reports

In addition to these, the following management reports are also provided to the state level staff through the CRS:

- Number of Yearly FTE Teaching Faculty
- Average Student Credit Hours per Yearly FTE Teaching Faculty
- Yearly EFT Students - Yearly Teaching Faculty Ratio
- Student Credit Hours Produced and FTE Teaching Faculty
- Building Inventory Report (Inventory of College and University Physical Facilities [OE Form 2300 - 7] Report as required by the Department of Health, Education and Welfare)
- Facilities Inventory Edit

Clearing-house Function

At this university the institutional research and planning office has been designated to transmit the bulk of the data to the state board although this office neither generates nor maintains any of the data required for the reports. The office performs a clearing-house function, instead (Bryson, 1977). Analytical analyses regularly performed by institutional researchers on university data, apart from the state system, provide insight into the interdependence of data elements for valid and reliable analytical work. The CRS, although externally imposed, has demanded some degree of internal consistency in the operational aspects at the institutional levels. Although the computerized records of each major area are utilized in the preparation of these reports, a tremendous amount of cross-checking and adjustment is necessary as a part of the clearing-house function to provide minimum basic data which can be used by the board. The extent of this effort demonstrates the need for standardization and consistent record-keeping procedures to be further developed at the institutional level.

Institutional researchers have been instrumental in identifying weaknesses inherent in the state-level management information system stemming from inappropriate use of inadequately defined data originating in 32 separate institutions. Consequently, individuals from this office have been enlisted in an advisory capacity to the state-level governing board as they seek to establish data definitions and standardize procedures for interpreting the data. Particular emphasis has been focused on faculty workload and curriculum information.

Organizational Data Flow

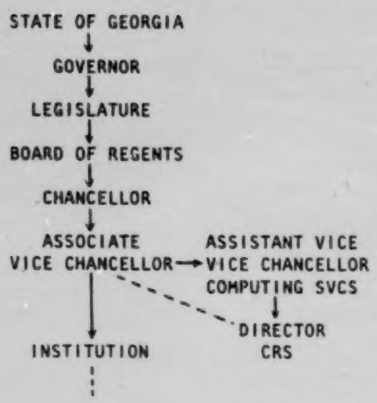
In considering the organizational data flow to a statewide management information system, one should also take into account the organizational structure of heterogeneous postsecondary institutions and the state system. The organizational charts of institutions vary widely. This is because these organizations have been developed to reinforce the goals and values of the institutions and support the leadership styles of the presidents. Organizational charts are frequently indicators of the decision-making structure. This is an important consideration because the success or failure of management information systems hinges on their capacity to be integrated into the decision-making process (Schroeder and Adams, 1976).

Chart 2 illustrates the organization of higher education in Georgia as it relates to the statewide management system.

The organization at this university shows the vertical flow down from the President to the Executive Vice President and then to his staff support office of institutional research and planning. From this point, the organizational structure

CHART 2

ORGANIZATION OF HIGHER EDUCATION IN THE STATE OF GEORGIA


STATE OF GEORGIA
GOVERNOR
LEGISLATURE
BOARD OF REGENTS
CHANCELLOR
ASSOCIATE VICE CHANCELLOR
ASSISTANT VICE VICE CHANCELLOR COMPUTING SVCS
DIRECTOR CRS
INSTITUTION

for data flow to the statewide management system is horizontal rather than vertical. It criss-crosses many vice presidential areas and intertwines numerous subordinate offices. Table 1 illustrates the interdependence and interaction of the five subsystems of the CRS on the organizational entities at this particular institution.

TABLE 1

INSTITUTIONAL DATA FLOW

OFFICE/AREA	CRS				
	PIR	FAR	CIR	FIR	RUR
PRESIDENT					
EXECUTIVE VP		0			
VP - ACADEMICS		0			
ACADEMIC DEANS (6)		0			0
ACADEMIC DIRECTORS (2)		0			0
DEPARTMENT CHAIRMEN (54)		0			0
ACADEMIC DIRECTORS (2)		0			0
LIBRARIAN		0			
VP - FINANCE		0			
TREASURER					
PAYROLL	0				
COMPTROLLER					
PERSONNEL	0				
ADMIN SVCS	E				
VP - STUDENT LIFE		0			
REGISTRAR			0		0
ASSOC VP - ACAD SVCS		0			
ACAD SVCS	ET				
ASST VP - IR & PLANNING, & INSTITUTIONAL COORDINATOR CRS		0			
DIR SPACE UTILIZATION (REPORTS COORDINATOR)	C	CET	CET	OET C	OET C

0 - data origination
E - editing & content monitoring
C - coordination
T-- transmits to state

Chart 3 demonstrates how the data flow of one of the sub-systems is accomplished as input to the CRS. Note that the only end product for the institution is the error report.

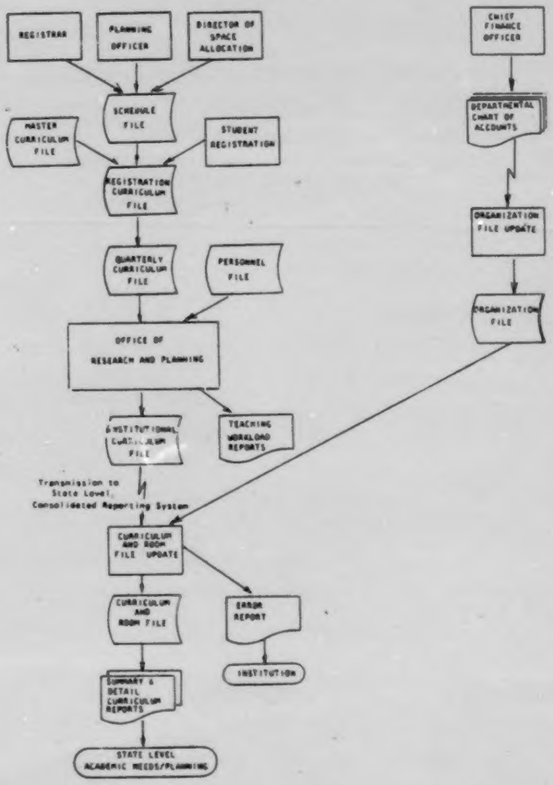
Perceived Improvements Needed

The need for concise, timely, consistent and relevant information is ever present in an institution of higher learning. The statewide management information system was expected to serve the needs at various levels, i.e., institutions, governing board, legislature, governor, as well as federal and various non-government entities. In our opinion, neither the impact of this management reporting system upon the decision-making structure of this institution nor its benefits in evaluation and long-range planning has been realized to date. A tremendous amount of raw data has flowed upward to the state level, yet scant information has flowed downward to the institutional level in return. Summary and comparative information compiled at the state level, as yet, has not been received within an adequate time frame for institutional management applications.

A two-way communications link between information systems (either at the state level or within the institution) and the institutional decision-making process is of paramount importance. Potential implementation of other statewide data systems applications raise many pertinent questions relative to this issue. Will the institution receive analytical summary reports drawn from the data submitted to the state board? Will the institution be limited only to certain

CHART 3

CURRICULUM INVENTORY REPORT


REGISTRAR
PLANNING OFFICER
DIRECTOR OF SPACE ALLOCATION
CHIEF FINANCE OFFICER
SCHEDULE FILE
MASTER CURRICULUM FILE
STUDENT REGISTRATION
DEPARTMENTAL CHART OF ACCOUNTS
REGISTRATION CURRICULUM FILE
ORGANIZATION FILE UPDATE
QUARTERLY CURRICULUM FILE
PERSONNEL FILE
ORGANIZATION FILE
OFFICE OF RESEARCH AND PLANNING
INSTITUTIONAL CURRICULUM FILE
TEACHING WORKLOAD REPORTS
Transmission to State Level, Consolidated Reporting System
CURRICULUM AND ROOM FILE UPDATE
CURRICULUM AND ROOM FILE
ERROR REPORT
INSTITUTION
SUMMARY & DETAIL CURRICULUM REPORTS
STATE LEVEL ACADEMIC NEEDS/PLANNING

standardized reports? Questions regarding the extra staffing and computer hardware required by the institutions to meet future demands need answering. It is expected that under a statewide centralized personnel/payroll system that institutions would not be permitted to maintain a local data base to support institutional needs but would transmit all payroll/personnel data to the state office. If the same philosophy and procedures were incorporated into the other centralized systems such as the student records and budget systems, it is likely that the local institution would be at a great disadvantage to perform many of the analytical tasks so necessary for day to day operations and effective management at the local level. During the preliminary design phase of these centralized systems, there has been little exploration of the impact of such undertakings upon the operation of the individual institution. Although institutions have opportunity for input into task force activities, it is unlikely that a single institutional representative could foresee the ramifications of statewide systems upon the many operational areas of an institution. The level of the state board's commitment to the implementation of these centralized systems is already evidenced by the investment of funds into massive software and hardware at one installation.

Centralized, computerized information-gathering systems are powerful managerial tools which can be of benefit to higher management, but they can also subvert the efficiency of the lower level management function if the information is not continuously shared with those officials directly responsible for institutional operations. One step in the right direction would be to share the results of the present CRS with institutions on a quarterly and timely basis. Timing is essential, and institutions need prompt feedback so that our operations can be evaluated within the perspective of the performance of the state system as a whole, and so that prompt action can be taken immediately. At present, summary information based on the CRS is only received many months after the end of the fiscal year. It should be observed that prior to this report dissemination, institutions are required to furnish an annual report of similar information apart from the CRS.

Discussion

An elegant management information system may look impressive, but it is impractical if the institution cannot obtain operational data or meaningful analytical information. Data element dictionaries may clarify data needs, but they cannot guarantee that the data will exist. Some data flow questions sound so simple that one may find it hard to believe that they represent problems; but meshing them on a complex system is no simple feat. The system may appear to have great potential for analysis, but unless enough attention is paid to all the details inherent in the data, it may be incapable of producing consistent reports.

It is obvious that statewide management and reporting systems have the capacity to capture a lot of numbers, but are they supplying and satisfying the internal needs of the individual institutions on a timely basis? For a statewide data collection system to be successful, the primary consideration should be the internal management procedures of individual institutions and secondary consideration should be given to external needs. Further, these systems need to be incorporated into the decision-making process at both the state and local levels if the effort and expense are to be justified. Practical considerations suggest the development of a procedure by which the institution could assess the nature of its internal decision-making questions first, and then be given sound advice on setting priorities within an information system concept to address its own needs. The results might not be as spectacular as a statewide management approach, but it does seem likely to be more effective. Sobczak (1977) sums it up very well: "Data base is a systems philosophy, not a technology ... a people problem, not a software tool" (p. 150).

References

Bryson, C.H. The institutional research office: report factory or information clearing house. In Research and Planning for Higher Education, Proceeding of the 17th Annual Forum of the Association for Institutional Research, Montreal, 1977.

Laughlin, J.S. Statewide data collection for decision making: The Kansas scene, fact, fiction, fraction, frustration or fruition? In Information for Decisions in Postsecondary Education, Proceedings of the 15th Annual Forum of the Association for Institutional Research, St. Louis, 1975.

Perkins, J.A. The University as an Organization. N.Y.: McGraw-Hill Book Co., 1973.

Sobczak, J.J. A data base story. Datamation, 1977, 23 (9), 139, 142, 147, 150.

Schroeder, R.G. & Adams, C.R. The effective use of management science in university administration. Review of Educational Research, Volume 46, No. 1, 1976.

ONE SCHOOL - ONE COMPUTER
A SECONDARY SCHOOL'S MINI-COMPUTER APPLICATION

Don Caskey
Marist School
Atlanta, Georgia

In the five years since Marist School acquired its mini-computer system, mini- and micro-computer systems have grown in sophistication as their price has decreased. In that same time period, applications of computers in education have matured significantly. Because of these changes, schools which have heretofore found computer acquisition too expensive may now find the prospect to be an attractive way to deal with growing educational demands and spiralling operational costs. Drawing from the Marist experience, this paper describes for these schools a rationale for the acquisition of a multi-purpose, time-sharing, mini-computer system by a small school and some applications currently under way at such an installation.

Introduction

This paper is directed to three audiences. The educator in a small school or college who is unfamiliar with computer applications can gain some insights and references. The computer professional who is unfamiliar with educational computer applications will see a small school administrator's ideas and applications. The educator who is familiar with computers will find a configuration which, though not rare, is still the exception in present-day computer applications, namely one school (not a district or consortium, but a single middle/secondary school of about 800 students) utilizing a single medium-sized mini-computer system for a variety of administrative and instructional applications. Because of the experience from which this paper is drawn, it has some bias toward the secondary school application. It is hoped, however, that personnel of any small school (college, elementary, or middle or high school) could gain some ideas from this discussion.

Background

The computer is not new to educational institutions. Quite the contrary, it originated on the college campus (circa 1943).[1] Many of us who attended school 15 to 25 years ago can recall that, even then, only a decade after its birth, the computer center was a permanent fixture on the large college campus. That is still true, and not surprisingly so. With the cost for one year of higher education close to $3,000 per student (higher for graduate students), it is reasonable to assume that large campuses would quickly attain the economies of scale and financial base necessary to support a sophisticated computer installation. Furthermore, computers have become necessary tools for research, as well as for financial and student-records accounting.

Following the example of universities, many large school districts have, in the last decade and a half, centralized such operations as financial and student-records accounting, thereby gaining the purchasing power to support a district computer system. These systems have even been expanded to provide computer services at the instructional level. To deal with the decentralized nature of a large school system, instructional applications have been implemented on time-sharing[2] computers, where computer terminals within the schools transmit and receive data via telephone lines to a central computer.[3]

The Small Campus

But the above description pertains to large colleges and school systems. What about the small campuses of say 500 to 2,500 students where per-student costs can quickly become prohibitive and funds for capital outlay are very limited? There are a great many such schools across the nation. For example, according to the *1976-77 Standard Education Almanac*, there are nearly 30,000 secondary schools in the United States, with total enrollments in excess of 15.6 million.[4] This means that, on the average, a secondary school has only about 525 students. However, secondary schools are usually part of a large multi-secondary school system which can afford a computer system--right? Wrong. According to the *Almanac*, there are nearly 16,130 school districts in the country[5], and, with many of the larger metropolitan districts encompassing many secondary schools, we must conclude that most districts have only one or two secondary schools, a figure which has traditionally been much too low to justify the acquisition of a computer system.

In addition to the secondary schools, there are a host of other small schools. Nearly 1,800 of the 2,600 four-year and two-year colleges in the United States have enrollments less than 3,000, and almost 1,000 of them have enrollments below 1,000.[6] There are also over 77,000 elementary schools[7] and many hundreds of business and trade schools, most in the below-2,500 category.

What does the future hold for these thousands of educational institutions? According to the report in the March 1977 *AEDS Bulletin*, it holds a 10% decline in enrollments and a doubling of per-pupil costs (from $1,446 to $3,000) in the next six to eight years.[8] With decreasing enrollments and skyrocketing expenses, these schools certainly cannot be expected to launch expensive and costly computer innovations. On the other hand, if computers are more affordable for the small campus now than in the past, they could offer relief from these pressures. But are they affordable? The answer to this question might be found by answering two others: "What can computers do for the small school?" and "What do they cost?"

What Can Computers Do?

Firstly, they can do in education what they have done for many years in other industries, namely accounting. These applications may range from financial accounting (e.g., payroll, general ledger, receivables, and payables reports) to student-records accounting (e.g., scheduling, report cards, label and roster printing).[9]

Applications in the Guidance Office are gaining popularity. Guidance counselors have found that, with a computer terminal accessing a large data base, they can assist students in exploring many alternative college selections or vocational choices. With both private and public data bases available, this is one of the faster-growing computer applications today.[10] In the area of instructional applications, there are several approaches, all of which have been thoroughly tested for many years. Because of the expense of operations and the immaturity of this new educational method (still less than 15 years old) and the inherent slowness of the changing educational pedagogy, many of these methods are only today becoming practical. Briefly, they are: 1-Computer literacy (i.e., teaching students the capabilities and limitations of computers).[11] 2-Use of the computer as a tool (an important functional and college-preparatory skill similar to typing). 3-Using the computer as a student. To demonstrate and develop his knowledge of the subject, the student is asked to teach the computer to solve problems.[12] 4-The computer can perform lifelike simulations of experiences too expensive or complex for students to actually conduct--for example, economics students can engage in business competition, physics students can perform a Millikan oil-drop experiment, and biology students can simulate a fruit-fly genetics experiment at the computer terminal with all factors, including random chance, incorporated in the program. 5-Computer Assisted Instruction (CAI), the method whereby actual course materials are presented to the student at the computer terminal.[13] 6-Computer Managed Instruction (CMI), a relative of CAI, where the emphasis is shifted from presentation of text material via the computer and toward record keeping, evaluation, and interactive movement of the student through a set of learning experiences suited to him individually.[14]

What Do They Cost?

That the above computer applications have been developed with computers still selling at relatively high prices is an indication of how cost-effective and educationally-valuable such applications can be. But what is exciting is this: Indications are that computers are likely to continue to increase in power and storage while decreasing in price for the foreseeable future. If this is so, the small school may be about to enter a new era, an era in which it will be not only economical and desirable to have an on-site computer system, but as necessary as having typewriters.

The simple fact is that the same thing is happening to the cost and quality of computers as has happened to calculators over the past few years. For example, Charles Lecht, in a 1977 *Computerworld* article, states that "the cost of performing 100,000 computations has fallen from approximately $1.26 in 1952 to roughly five cents in 1970 and about one cent in 1974. By 1977, this incredible improvement in cost performance, is expected to approach .5 cent per 100,000 computations."[15] By way of comparison, if the same advance in cost performance had been made in the auto industry, a new car today would cost about $20.00, instead of $5,000. What this means to the small school is that a powerful medium-sized computer system which marketed ten years ago for a half million dollars can be gotten today for between $20,000-$80,000. (A used system such as Marist's sells for $30,000 to $45,000.) What's equally impressive about these technological advances is that the end is not yet in sight. New major breakthroughs in computer development will continue to increase the power of computers and decrease their cost. Such hardware advances as semi-conductor memory, bubble and charge-coupled storage devices, and micro-processor chips are only, within the last year or two, finding their way to the market. Most professional publications predict that these advances will have a major impact on costs and performance of mini-computer systems in the next few years.

In addition to these improvements which effect the computer itself, major advances have been made in development of low-cost, high-speed devices which "talk" to the computer, such as printers and CRT (video) terminals. Terminals capable of accessing time-sharing computer systems can now be purchased for as little as $300, with $1,000 to $1,400 buying a fairly sophisticated such device.

As if these hardware advances weren't enough, a major concept in the use of the computer has emerged in the last few years. The last such major innovation in computers was the invention of time-sharing, the concept mentioned earlier which allows many users to access a computer. This concept is called Distributed Processing, and it means that a network of several computers can now communicate with one or more larger computers, much the way terminals communicate with a time-sharing computer. The implication of this for the small school is that its mini-computer system may access a nearby larger computer system (perhaps owned by a school district or nearby university) to handle jobs which might require more processing power or storage space than is affordable for the small school.[16]

What can a small school expect to pay for its computer applications? There are too many applications and too many ways of evaluating costs of operations to present a firm figure. Typical costs, however, seem to range anywhere from $2.00 per student to $50.00 per student, including acquisition, maintenance, and software. If the computer has demonstrated value in administrative, financial, guidance, and instructional applications, then even the $50.00 figure would represent a modest portion of the almost $1,500 per student which is currently being spent.

The One-school, One-computer Idea

Of course, there are other alternatives available to schools than the purchase of a single multi-purpose computer system. A school can time-share on a large computer system, the configuration described earlier in this article, or it can purchase more than one computer system. The first alternative, though still widely used, seems to be becoming less and less practical, as more terminals are needed within a given school. A primary reason for this diminishing practicality is the increasing communications costs which invariably accompany an increase in the number of ports. In a recent article in Technological Horizons in Education, S. Charp and H. Altschuler stated that, along with decreasing budgets and increasing personnel costs, "continuing increases in the cost of communication has hindered the possibility of large computer (time-sharing) networks.."[17] Since much of a small-school operation is predicated on a multiplicity of terminals being available (i.e., a decentralized computer operation without a "computer center and accompanying personnel"), communication costs, usually in the form of telephone bills, can quickly become exorbitant, outweighing, for even modest applications, the cost of acquiring an in-house mini-computer system. Furthermore, there is evidence that larger computer systems are not inherently more efficient than small computer systems.[18]

At the opposite end of the spectrum is the current boom in micro-computers, sometimes known as hobby computers. It might be reasonable to assume that two or more micro-based computers could be acquired to serve independent applications, thereby separating, for example, administrative operations from instructional. This is being tried more and more. However, problems can arise. Supervision and maintenance of several systems usually costs substantially more than for one system. Similarly, upgrading several systems (and, with the boom in computers, technological obsolescence occurs quickly) is frequently more expensive than a single system. Also, it may be noted that, given the economic/political considerations of a small school, it is common for priorities to be set on which system gets the upgrade. Since the cost-effectiveness can usually be more easily established for the administrative computer, instructional computer applications sometimes suffer. Another factor is that, having divided the available resources, a school may find that the money left for the administrative computer will not buy a machine powerful enough or with sufficient storage to handle the demands of typical administrative activities. It is also conceivable that CMI/CAI applications could be found in a similar situation. Sharing a single mini-computer system can provide large capacity and storage at various times for several applications. Even given the above factors, the micro-based system has an apparently overwhelming cost advantage over the medium-sized mini-computer system. But it must be assumed that a small school does not wish to add full-time programmers, and the fact is that current micro-computers do not have as sophisticated language and hardware capabilities as their larger brothers. According to

Edward Yasaki in a 1977 Datamation article, a "typical micro-based system...might consist of $7,000 to $10,000 in hardware, considerably lower than the $20,000 mini- computer-based system." But, he states, "It's not going to take...long before the extra problems associated with programming such a primitive machine will consume the difference in price."[19] Reiterating the point, Esther Surden of the staff of Computerworld summarized the opinions expressed at the 1977 National Computer Conference by stating "micro-computers have a long way to go before they are the user-oriented system they would like to see."[20]

At Marist: The Computer Hardware

A common scene in many science- fiction epics is that of the hero pushing back the rubble of a destroyed civilization, only to uncover an ancient computer system still operating without human assistance after many years of neglect. No, the state-of-the-art in computer science is not to that stage yet; however, the Marist computer, an HP2000F time-sharing system, does spend 23 of its 24 hours each day alone, purring away quietly in a dark closet-sized room.

One of the primary considerations when the system was purchased was to avoid the expense of additional staff members for computer operation. It was found that this mini-computer system came very close to being operator-free. For one hour a day, a student assistant performs routine maintenance on the system, such as making magnetic-tape copies of essential data, entering new I.Ds., mounting requested files from mag-tape libraries, etc. All other access to the computer is handled via time-sharing terminals (system capacity - 32) located in various parts of the school building. High-speed printing facilities used for administrative tasks consist of a high-speed terminal (GE TermiNet located in the main office). Since 95% of the administrative reports and printouts use the same-sized paper (blank 8½" x 11" sheets), forms mounting, a common computer-operator task, is virtually non-existent.

The Software

An invaluable asset to many college computer operations is the programming services of volunteer students. To a limited extent, high schools can utilize the same services. This seems an extraordinary claim, but the fact is that better than 50% of the programs utilized for administrative operations at Marist were actually coded by students. As long as the programming tasks are designed by the supervising faculty member and kept reasonably modular, students in levels as low as grade eight have been able to make significant contributions. Though some of the older programs are unstructured and internally amateurish, they still work sufficiently well that the package of administrative programs has been marketed outside the school. Also in use at the school are CAI/CMI and file-management packages which were furnished at no cost by the manufacturer when the system was purchased. The GIS system mentioned earlier is also leased for use in the guidance office.

The Math. Lab

Room #217 was an ordinary classroom until, about five years ago, the Math. Department claimed it as the Math. Lab. Then it became a resource and tutorial center for students with needs for remedial help or enrichment or with some time to kill. The most important feature of the lab was the full-time certified staff member who proctored and tutored therein (how the Math. Department pulled this off is another story). This was about the same time that the school acquired its mini-computer system, so, since the Math. Department would certainly be a major user of the computer, computer terminals were put in the Math. Lab. At that time, the terminals consisted of an obsolete CRT (video terminal) which had been purchased with the computer and a model 33 teletype, left over from a two-year time-sharing experiment with a nearby university. Today, these facilities have grown to two teletypes, two ADDS 580 CRTs, two TermiNet 300 printing terminals, a Digilog 103 CRT, and a TI 700 printing terminal.

But the terminals in the Math. Lab are not exclusively for use by the Math. Department. On these terminals, at any given time, students may be running any of the following: Experiment simulations for a chemistry, biology, or physics class; remedial CAI math. lessons or math. drill-and-practice quizzes; CAI Spanish lessons; statistics or calculus problems for those courses; a prefix-suffix drill which is required by the English Department for students in the required vocabulary course; logic or matrix programs for algebra classes; the students' own programs for one of the two quarter-long computer-programming courses (BASIC and beginning FORTRAN); or, finally--that's right--computer games such as Star Trek. Students also use the lab to write a great many programs for the faculty and student organizations, including programs to help with the student council elections, departmental inventories, teachers' grades, and sports statistics. The terminals in the lab are rarely idle.

"Library Book Policy" on Terminals

Nor do all of the Math. Lab terminals fall idle at night. Two of the terminals, the TI and the Digilog, are portable and are checked out to students or faculty in the same manner as a library book. The user then calls the computer over his home phone and works into the evening uninterrupted. Of course, the computer must also have phone lines to do this, and they can be expensive. Marist helps defray this expense by permitting other schools to time-share on the system, using the same telephone lines, during the day. In the past year, three colleges, three high schools, and an independent guidance center have time-shared on the Marist system. However, demand for such services is quickly shrinking as communications costs increase and computers become less expensive. So, it is recommended that schools which are acquiring computers should not plan on such financial assistance.

The Guidance Office Applications

A terminal (soon to be two) is permanently available in the Guidance Office for a variety of applications. Guidance personnel use it to query the student-records' data base for grades and behaviorial data, in preparation for counselling sessions with students and/or parents. The Guidance Office asks every junior and senior to log on sometime during the year to the Guidance Information System for the purpose of exploring college alternatives. A video-tape is available in the library to instruct students in the proper use of the GIS package. An on-line registration system can be made available to this terminal and the Math. Lab terminals for the purpose of having students enter their course requests for the coming year. Students working at this terminal have the advantage that a counselor is nearby for advice.

Administrative Activities

All course requests, scheduling, grades, and permanent transcripts are handled via a large data file on the system. Three CRT Terminals and one Termi-Net 1200 high-speed printing terminal are used in the main office for inquiry to this data base. A software package of 200 programs called MARAD (mentioned earlier) was written at Marist for managing the various labels, reports, rosters, and special listings associated with these tasks. The data base is always available from the time-share terminals (security measures prevent other terminals from modifying the data base) via the software package's BASIC programs. For more information about Marist's MARAD package, see the January/February 1978 Hewlett-Packard Newsletter.

Finance and Alumni

Marist's Alumni Office also keeps a data file on the thousands of graduates of the 78-year-old institution, for the purpose of mailings, meetings, and fund drives. Also, the Finance Office uses the system for its Accounts Receivable activities and, soon, a budgetary accounting system. A CRT is available in the Treasurer's Office for this latter application. These are sort of "bread-and-butter" activities, which can be readily shown to be cost-effective. Since they are fairly common, a detailed explanation of the nature of these activities will not be given.

Concluding Remarks

Five years ago, Marist bought a mini-computer system which, at the time, seemed larger than its needs required. Twenty-three-million characters of storage and 80,000 characters of computer memory seemed to have unlimited potential. Today, with over 20 ports active throughout the building, the school has found that the computer system permitted exploration in many areas not yet imagined in 1972. With the increased opportunities for computer applications in instructional and guidance areas, it now appears as if this 32-port mini-computer system, far from being extravagant, is a very good size for a single-school application.

Footnotes and Bibliography

[1] A. Strauss, "ENIAC" The First Computer," Mathematics Teacher (Jan. 1976), p 67.
[2] See: R. Rusch, Man's Marvelous Computer (N.Y.: Simon & Schuster, 1970), p 30.
[3] e.g., see: D. King, "Developing a Metro Computer Consortium...," T.H.E. Journal (Dec. 1976), p 13.
[4] Standard Education Almanac (9th edition, Chicago, Ill.: Marquis Academic Media, 1976-77), p 249.
[5] Ibid., pp 249 & 246.
[6] Source: Guidance Information System (Timeshare Corporation, Hanover, N. H., Oct. 1977).
[7] Standard Education Almanac, loc. cit., p 249.
[8] "ICCP Calls For Help," AEDS Bulletin (March 1977), p 1.
[9] e.g., see: D. Caskey & T. Stokes, "MARAD," Hewlett-Packard Newsletter (Jan. 1978).
[10] e.g., see: Guidance Information System (see above) and the Career Information System of the University of Oregon.
[11] See: "NCTM Adopts Position Statements," Ga. Council of Teachers of Mathematics Reflections (Jan. 1977), p 7.
[12] The Computer Assisted Mathematics Program by Scott, Foresman & Co., 1969, is a good example of this approach.
[13] For CIA evaluation, read: W. Cunningham, "Technology: A New Era in Education?," AEDS Monitor (Jan. 1978), p 5.
[14] The Hewlett-Packard Newsletters from 1976 on are good references for CMI.
[15] C. Lecht, "The Waves of Change," Computerworld (April 18, 1977), p 10.
[16] F. Vaughan, "Technology Credited as Spark of Distributed DP Revolution," Computerworld (May 9, 1977).
[17] S. Charp & H. Altschuler, "1966-1976 A Decade of Usage of the Computers for Instructional Purposes, T.H.E. Journal (Nov. 1976), p 10.

[18] C. Reynolds, "Issues in Centralization," Datamation (Mar. 1977), pp 91-98.

[19] E. Yasaki, "Microcomputers: For Fun & Profit?," Datamation (July 1977), p 68.

[20] E. Surden, "User-Oriented Micros Seen in Distance," Computerworld (June 27, 1977).

A COORDINATED MIS APPROACH TO COMPUTERIZED ACCOUNTING SYSTEMS DESIGN

James R. Cook
Marvin A. Johnston
Michael C. Gallagher
Southwest Texas State University

In the next few years, it will be unusual to find an organization of any size that does not rely on electronic data processing in some form. The authors feel that the true potential of computers has not been fully realized and that its role in the area of decision analysis will become more pronounced in the next few years. Thus, the door to the computer's true profit potential has not been unlocked. (3:17-31) However, when optimal benefit from utilizing a computer is reached, industry, government, and business are going to look to education to supply the people with the know-how who can help them achieve these "potential benefits." Therefore, a tremendous responsibility rests on the shoulders of educators to provide students in computer science with the knowledge to actually get the job done in the real world.

Theoretical concepts are an important facet of education, but there comes a time when theoretical concepts must be adapted to some practical purpose. Modern-day university students are demanding, and rightly so, that universities provide them with knowledge that will help them be able to actually "do" something when they take a position of employment. Therefore, it is the responsibility and duty of educators to develop the type of curriculum which will satisfy the needs of the students. We as faculty of colleges and universities exist for one primary purpose and that is to educate and train students so that they may make some useful contribution in the real world. It may be possible for graduates in some disciplines to "fake" their way through their job responsibilities. However, students of computer science are revealed for what they are if they do not have the knowledge to actually produce. The reason for this is that computer science is a highly technical field and a highly competitive field.

Many educators feel that computer science or data processing does not belong in a college or university. They believe that computer science courses should only be taught in trade schools. One reason for this belief is that some individuals have the impression that computer science merely involves programming. These people do not realize that computer science involves much more than just learning about bits and bytes. This is especially true in schools of business.

The complexity of our present day society confronts business, industry, and government with an ever increasing amount of information. This information is normally interrelated. However, the volume of data and the intricacy of relationships certainly exceeds a human capacity to relate the many factors that are necessary to make decisions.

Approximately eighty percent of the usage of computers in our society is in the commercial area. Computer science graduates working in the commercial area must be involved with people and problems from all areas of business. Thus, business students in the computer science area must have some working knowledge of accounting, finance, marketing, management, and economics. Computer science graduates in the commercial area must not only have knowledge concerning the technical aspects of computers, but also must be knowledgeable in problem areas of business to which electronic computers will be applied. Therefore, educators in this area must be able to educate students such that they can enter an environment in which many and varied types of unstructured problems exist and be prepared to attack these types of problems. A business student with a computer background and knowledge of all areas of business is definitely a most valuable asset to an organization. This type of individual is not just a programmer, but is a "thinker" and a "problem solver" who is able to attack unstructured types of problems as opposed to structured types of problems often encountered in classrooms and in textbooks. The authors in day-to-day dealings with executives of business firms have learned that they feel we are finally beginning to give students the type of curriculum that they need and one that is desired in the business world.

Prior to the past few years, firms were forced to train college graduates before they could work in the area of decision analysis or in computer-related areas. In the past few years, some very fine computer science programs have been developed in schools of business.

The computer is here to stay. It is an expensive tool to keep on hand, and management is demanding that computers be used in ways that will improve the profit position of the firm. Business leaders are beginning to see some of the potential areas of success involving utilization of computers. Many firms are holding seminars concerning the position and importance of the computer within the firm. Many employees are going back to school to learn about computers and how to use them. Computer science programs, such as the one described here, are attempting to train college graduates for the real world. These graduates will have had education in all areas of business and in addition are versed in computer programming, statistics, operations research, and information systems. The objective of this type of program is to provide firms with technically competent individuals who can attack complex, unstructured types of problems and thus give management the guidance to make quick and sound decisions. Such a program, as mentioned above, exists at Southwest Texas State University and is entitled "Management Information Systems." Areas of decisions analysis that are included in this area are computer programming languages, file structures (for purposes of data base development), information systems, operations research, statistics, and computer systems development and design. The degree program in Management Information Systems (hereafter referred to as MIS) is not a 12 hour or 15 hour concentration in data processing or computer science as is present in a number of universities. The degree program consists of a full 36 hour curriculum. In addition to the curriculum in MIS, the student is required to take coursework in the areas of accounting, management, marketing, finance, and economics. With the widespread background in business and his technical knowledge in areas such as programming, file structures, and information systems, the MIS graduate can assume his responsibilities in business and industry and make a definite contribution to the firm almost immediately.

Those of us in academia sometimes fail to keep abreast of what the current trends and problems are in the real world. In some instances this is not the case, but instead academicians never really know what is happening in industry and business. Educators many times are engrossed in their research endeavors or in their "pet" models or theories rather than teaching the students and thus providing these graduates with the knowledge and skills that will enable them to make a useful contribution to business, industry, or government in the area of decision analysis.

The MIS degree program at Southwest Texas State University attempts to give business students in the computer science area the knowledge and the technical skills to help provide management with the counsel that will help them make quick and sound decisions. The students from this degree program will be adept in using programming languages, operations research tools, statistical tools, and concepts of file design and data base management. But even more important, the MIS graduate is flexible and thus can adapt his thinking and problem-solving abilities to the particular problem. The MIS graduate does not feel inclined to try to match some textbook type of problem or example to the problem he is trying to solve in the real world. The reason for this is that he has been taught to adapt his knowledge of all areas of business, his knowledge of programming, his education in information systems, and his education in systems analysis and design to "unstructured" types of problems rather than merely to match problems with "textbook solutions."

An example of how the student in the MIS program is exposed to problem areas of an unstructured nature is in a course entitled "Introduction to Management Information Systems." An extensive case study was assigned to the class in which the class was broken down into groups of from three to five students. One benefit of the project was that students had the experience of working with other individuals. The case was a thirty-page case which involved an actual firm and their financial and organization difficulties. A systems study was performed by each group. The studies involved analyzing the present condition of the firm, studying the responsibilities of key operating personnel, analyzing the value of present reports being generated, analyzing current systems and procedures, and forecasting future growth and needs of the firm. The students would then develop their solution to the problems at hand. This involved designing the various systems such as accounts receivable, sales analysis, and inventory. It also involved development and design of the data base, estimations of costs of all nature concerned with implementing the new system, and the selection of computer hardware and software. We have received numerous comments from students saying that this was one of the most meaningful and useful projects they have been associated with during their university coursework.

Another example of the type of curriculum existing in the MIS degree area is in a course entitled "Computer Systems Development and Design." This course is a senior level course. The students taking this course have had one or two semesters of COBOL programming, "Management Information Systems," and a course in file structures. Taught in this course are principles of systems analysis and design. However, the true value of this course lies in the application of these principles of systems analysis and design to actual problem areas which are in the form of class projects.

In typical programming classes, students are assigned programs to write involving problems of a "structured" nature. What is meant by "structured" problems in regards to computer programming classes is that the problem is narrowly defined for the student, the form of the input is defined, and the format of the output report is also specified or delineated for the student. All the student has to do is merely code the program. Therefore, the problem is completely "structured" for the student.

Class projects are employed in this course in order to give the students the opportunity to attack problems which are of an "unstructured" nature. Each student in "Computer Systems Development and Design" is assigned to work with a student enrolled in a course entitled "Accounting Systems," which is taught by the accounting department. Each accounting student has the responsibility for designing some type of system such as inventory, accounts payable, accounts receivable, or some other system.

Assume that the system to implement is an accounts receivable system. The accounting student must be able to communicate to the computer science student what his problem entails. The computer science student must learn what procedures are involved in handling and recording accounts receivable transactions, what types of information must be recorded and stored, how often accounts receivable are processed, what reports are required by management, how often these reports are required, and what the contents of these reports are.

It is then the responsibility of the computer science student to take the system which was designed by the accounting student and actually implement the system on the computer. The computer science student must design the files that will hold summary information on each customer, information concerning the actual transactions, and information concerning payments made by the customers. The student must also design input record formats and the format of output reports. The student would also consider such matters as the creation of the master file containing summary information of each customer, and the creation of the files containing information on the transactions and payments. The student must also consider the updating of these files as new transactions and payments occur each accounting period. In addition, consideration must be given to the handling of new customers who must be added to the files and old customers who may need to be deleted from the files. After consideration of all these matters and after extensive analysis and design of the systems, the programs are written to implement the system. A system such as the one described here would probably be programmed in COBOL and would by necessity employ the most sophisticated techniques and features of COBOL. Actual "test" data would be constructed by the accounting student for purposes of testing the system. This type of "unstructured" problem forces the computer science student to use his knowledge of functional areas of business, his knowledge of programming, his knowledge of file structures, and his knowledge of systems analysis and design in order to cope with a problem of an "unstructured" nature.

A detailed explanation of a specific accounting systems design may be found in the appendix. This project involved a small business located in South Texas.

In summary, the authors feel that educators in computer science and decision analysis should reexamine their objectives and their curriculum in terms of what type of graduate we are producing. Are the graduates leaving colleges and universities today able to make a useful contribution and impact on business, industry, and government or are they merely lost and out of place in the real world?

LIST OF REFERENCES

1. Early, G. G., Johnston, Marvin A., and Poirot, J. L. "Information Processing: A Prepared Major at Southwest Texas State University", A Paper Presented to the Academy of Science, Houston, Texas, 1972.
2. McDonough, Adrian M. Information Economics and Management Systems, McGraw-Hill Book Co., New York, 1963.
3. "Unlocking the Computer's Profit Potential", McKinsey and Company, Inc., McKinsey Quarterly, pp. 17-31, Fall, 1968.

APPENDIX

To fulfill the "unstructured" problem requirement for "Computer Systems Development and Design," a lumber company in South Texas was selected for study. After several interviews with persons responsible for operations of the company, it was decided that the company's accounts receivable system would be restructured and computerized.

The new system would generate three reports required by management: (1) an accounts receivable ledger, (2) a sales summary report, and (3) a customer address listing.

The accounts receivable ledger would itemize payments from and purchases by each customer. Purchases (both cash and credit) listed on the ledger would also be accompanied by an alphabetic description of the type of purchase (i.e., lumber, hardware, etc.). Beginning and ending credit balances would also be included on the ledger.

The sales summary report would break down purchases by category (i.e., lumber, hardware, etc.). Total dollar amounts and percentages for each category would be calculated and printed along with a grand total for gross sales.

The customer address listing would be alphabetized by customer name and would contain the account number and current credit balance for each credit customer.

All processing would be in batch mode (given the size of the company) and would be performed by an independent data processing company. Two programs (COBOL coded) were written to handle the processing. One program would update records when necessary (i.e., address changes). The other program would perform the posting to the various accounts. Transactions would be input in sequential fashion, processed and credit balance changes would be posted to the ISAM master file. Only transaction records containing bad data or a non-existent account number would be flagged and printed as a control measure.

ADP PRODUCTION TECHNICIANS:
NECESSARY PERSONNEL FOR SMALL DP CENTERS

Judith C. Ewart
College of William and Mary

Utilization of ADP personnel is one of the toughest problems that must be tackled by computer center management. Analysts want to continue programming; programmers want to run the systems they have written. There comes a time, however, when programmers spend so much time running existing systems that precious little time is left for new development work. When this happens, computer center management often feels the pressure to justify more programmer positions. An alternative solution - one which I would like to present in this paper - is that of establishing ADP Production Technicians positions to remove the responsibility of running existing systems from programmers so that their time can be spent in the writing of new software or upgrading of old software. While this separation of duties stems partially from a belief that control can be enhanced, the basic argument for the utilization of production technicians is budget relief.

"In the beginning" there were tab shops where data was punched, verified, sorted, listed, and tabulated. The famous IBM 1401 performed miracles; much manual and clerical effort was now "computerized." Such systems necessarily copied the manual systems and the 1401 was a fancy typewriter with form control and limited calculation abilities. Peripheral equipment - sorters, collators, reproducers, interpreters - did much of the work of preparing card data for final use on the 1401. Small shops, set up to economically handle accounting, billing, student grades, class rolls, et cetera, multiplied repidly accross the country.

Computer manufacturers suddenly realized the great wealth to be found in administrative applications. Machines were designed and marketed to more effectively handle administrative data. Tab shop managers and technicians were trained by vendors to be programmers and analysts. Often the vendors and company employees learned together in this new world of automated data processing.

Trained programmers and analysts were a scarcity. Only large organizations and universities could compete with the government for the few highly trained ADP personnel. Small companies and educational institutions, where money is almost always tight, were forced into a make-shift roll with questionably trained ADP personnel and totally uninitiated users. The profession was new and only a few made a groping attempt at standardization. Besides, the ADP employee was lucky to stay caught up with day-to-day demands and problems, much less have time to spend in personal education and training. Analysts and programmers, desiring to do a good job in a sometimes hostile environment, went overboard to try to solve everybody's problems and lessen everyone's workload - even if it meant that his personnel did the work themselves. Data entry staffs grew to accomodate the demand for data preparation. Programming staffs grew to accomodate system demands, even when unreasonable.

Programmers were known as "jacks-of-all-trades." They performed systems analysis tasks, wrote, tested and debugged programs, often operating the computer himself, and then became in charge of the working system. Normally one person was totally in charge of a particular system or multiple systems and little interaction existed between programmers. Each programmer worked with a user office and little was known of his programming effectiveness or efficiency because all problems were handled by himself behind closed doors. If outputs were received on time, the user was satisfied, thus the manager was satisfied. The busiest programmer was the best - the most dedicated; managers seldom had time to investigate individual programs or systems or had the knowledge to compare that against any type of standard.

Suddenly there's a problem! User offices, used to the quick response of earlier days, complain that nothing new is being accomplished, more mistakes are being made in current systems and more deadlines are being missed. What's happened? Systems have been developed and written without standards. The proliferation of inconsistant data formats and rules are confusing to data entry personnel who turn over rapidly. As programmers begin leaving due to dissatisfaction, pressure, or outside demand for trained personnel, new programmers find little consistancy and less documentation. Back-tracking becomes necessary; one must learn the old before proceeding with the new. Maintenance demands are ever increasing as user office personnel and requirements change over time. Ill-conceived systems must be made to incorporate new requirements; stand alone systems must be made to "communicate" with each other. In the educational institution faculties and boards of visitors vote in new regulations without regard to data processing problems; state and federal agencies constantly place both new and changing demands for data upon the institution.

Data processing costs begin to soar as personnel are acquired to absorb the new demands. Suddenly it is no longer cost justified to hire more personnel. Other means of solving the problem must be found. Programmers attempt to write programs to produce "pre-punched" maintenance cards and master header assignment cards for large jobs that used to require coding and punching. Terminal entry programs are written to place data entry responsibilities in the user office. Soon the data entry personnel have slack periods but programmers are increasingly rushed. Solution: lay off data entry operators and hire more programmers? NO. Stop accepting new systems for development? NO. Retrain data entry operators as production technicians and free up programmer time for programming tasks? YES!

Sounds great! All the textbook personnel theories advocate this last approach - separation of duties, efficient use of highly paid personnel, low-status employee retraining, advancement and over-all morale boosting. Great theory but difficult to put into practice.

The Computer Center of the College of William and Mary faced such a problem about 3 years ago. I was selected to be the ADP Production Control Supervisor, charged with the following duties:

1. Responsible for production scheduling, timeliness and accuracy in all reports generated by the payroll, personnel, expense accounting, student accounts receivable, student admissions, matriculation and graduation, dormitory assignments, student directories and various other production systems.

2. Communications with all administrative users involving scheduling of data requirements, ascertaining changes and enhancements desired and reporting these to the Coordinator of Administrative Data Systems Group, and "problem-shooting" any discrepancies reported.

3. Supervision of four Production Control Technicians and five Data Entry Operators involving work assignments and schedules, training, leave scheduling, work assessments and evaluations and other personnel functions.

4. Initiation of necessary back-up procedures for all systems.

5. Design of standard request forms and logging and transmittal procedures. Design of standards for Production Control documentation and JCL run decks. Design of internal programming standards to facilitate production efforts. Initiation and supervision of program library facility for production programs.

Many problems were encountered almost simultaneously and are tightly interwoven:

1. Programmer Resentment

Programmers are rather unique individuals. They are (or feel they are) creative yet disciplined, precise yet sloppy (let the computer find my errors), and proud to the point of stubborness. Try explaining to a programmer that the system he has developed, written and nurtured all this time is going to suddenly be run by relatively untrained, former data entry operators! Try explaining to a programmer that the users will no longer be contacting him but a peer who recently received the promotion to Production Control Supervisor.

2. Lack of Documentation

One of the reasons for the change in duties is to free the programmer to do some programming. Alas, he is instead asked to expend much effort in preparing the documentation he personally never needed. Feelings of "If I have to write it all down, I might as well run it myself" are frequently expressed. To make the situation worse, the Production Control Supervisor is a former programmer with the same problem - he must document his own systems while taking on all these new duties. And needless to say, the production requests and requests for programming changes do not stop during this emotion-laden transition.

3. Standards for Documentation

The first documents prepared by programmers show a major problem - inconsistant format and content of documentation. Much documentation is program oriented - not system production oriented. The Production Control Supervisor, no more familiar with "production systems" than the other programmers, attempts to impose a set of documentation standards in the way of questions/answers to insure full and consistant information. Those programmers who were already cooperating become frustrated at their wasted effort; fuel is added to the fire by those indignant programmers who are rebelling at the entire reorganization.

4. Standards for JCL Decks

Since the Production Technicians are not yet versed in JCL, the Production Control Supervisor attempts to set standards to aid in the training process. Programmers see this as an attempt to stifle their originality. "If you can't accept what I have now, you won't get anything from me." "If your technicians can't learn it, tell them to go back to keypunching." Again more demand for programmer time is required to set systems up for the production group; this is currently time which would not have been required had the Production Control Group never come into existance.

5. Control of Programming and Testing

Programmers were used to testing programs whenever and however they chose; now changes and testing have to be both separate and apart from production and also timed correctly. Production source and load libraries are established. Testing libraries are established for programmer use; this means that full testing is necessary before the production occurs - no more testing on production data and working long hours to debug under "live" situations. Now the programmer's job directly impacts on the job of another person - the production technician. Check-out and check-in sheets, testing proceedure documents, and test results must be given to the Production Control Supervisor for approval; a former peer is now judging one's work and reporting problems to the supervisor of the data processing department.

6. Lack of Continuing Interest in the System

Once a programmer has documented his system and turned the system over to the Production Control Group, little contact is made until problems occur. All user communications are with Production Control and the programmer quickly loses interest as the human element of the job decreases. User appreciation is focused on the technician, but user and technician complaints focus on the programmer. Soon all communications between technicians and programmers appear negative. In addition programmers sometimes have little interest in adding new features developed in-house or purchased to systems they are no longer running personally. Why expend effort to improve something for someone who is always complaining? Thus formal requests for such upgrades sometimes have to be initiated by the Production Control Supervisor; this is received very negatively by a few of the programmers involved.

7. Loss of Expertise

Not all details concerning a system can be documented in a coherent manner; even those that can seem to change quicker than documentation can be updated. When a programmer runs his own system, he is intimately involved with the details. Once the system has been turned over to the Production Control Group, the programmer's interests move on to other tasks. Thus the user request "I need the same report I got last year at this time" suddenly becomes a problem; one-time, quick-and-dirty programs seem to return to life to haunt a Production Control Group.

Technicians become easily frustrated if programmers cannot immediately answer their most minute questions. They tend to forget that the programmer hasn't been in constant touch with the system for some time and that the user and Production Control Group have probably initiated changes of which the programmer is unaware. Complaints become more severe as the gap of knowledge widens over time.

8. Comparison of Programmer Talents

As is true in every organization, some programmers have more initiative and/or more ability than others. As the production technicians begin receiving systems from all programmers and comparing them for quality and ease of production, programmer differences become more public. Criticism becomes more vocal as some programmers keep an interest in their systems and seek to update them with new and more efficient features. Again, the Production Control Supervisor, once a programmer peer, must report such differences and difficulties to the Data Processing Manager to have conflicts resolved and systems improved.

Three years later some of these problems still exist and seem to be inherent in the system. Others have quietly faded away and still others were brutally battled before resolution. Our situation as it exists today:

I have been promoted to systems analyst. The former Data Entry Operator Supervisor is now the Production Control Supervisor, but at a lesser state title because of a lack of programming experience. This appears to be working well but would not have worked in the beginning. The necessity for formulation of the various standards mentioned required programming expertise. Also the acknowledgement of control status was more quickly given a fellow programmer. After 1½ years I had accomplished about all I could from that vantage point; the lead technician knew the systems and was ready to regain her supervisory role. Only as a systems analyst and supervisor of programmers could I instigate further improvements which I felt were so necessary. Also since no programmer was hired to fulfill the duties I had had as a programmer and acting analyst, the already small staff had operated one man short for too long. The Production Control Group is now 1 person short from its original configuration; that appears to be sufficient now that the organizational tasks are mostly completed.

Our ADP administrative management meetings now include the Data Processing Manager, the 2 Systems Analysts, the Production Control Supervisor, and the Data Entry Supervisor. Equal consideration is given to ideas from each person during the resolution of a problem or the formulation of a new concept or project. Thus Production Control has gained much in status and is officially considered an integral part of our organization.

Would we do it again? An overwhelming YES is the answer of data processing management, user offices and most programmers. There is still some resistance and some problems and probably always will be. Even with this remaining resistance, much has been gained by the reorganization:

1. We now have several technicians and a supervisor who are conversant with almost all production systems; production no longer comes to a halt because of programmer absence

2. Many systems have been significantly upgraded from their original state at the formation of the Production Control Group. This upgrading of systems is important for a smoothly running production group but is also important for programmer educational/growth and morale. Writing programs for someone else to run is a great incentive to do one's best!

3. Programmers do now have much more time for programming tasks and personal education/enrichment. Several programmers have truly stretched their horizons because they were not burdened with production tasks.

4. The users are more satisfied; with all production requests centralized in one group, better planning and coordination campus-wide is possible. The programming improvements spurred on by the reorganization have greatly benefitted the users resulting in a drop in errors and reruns and a shortening of turn-around.

5. Management can now feel confident that most situations are under control. We are now facing our first ADP audit with some apprehension; 3½ years ago, however, pure panic and bedlam would have been our reaction to a reported audit. After the preliminary audit, an auditor stated we were among the best controlled institutional ADP centers he had audited! Quite a compliment considering what it would have been like 3½ years ago.

6. Technicians are receiving valuable on-the-

job experience and training which will in some cases enable them to be promoted to programmer status as vacancies occur. And everyone knows the value of in-house promotions on morale, initiative and performance!

Would I do it again? While the task was emotionally and physically demanding and sometimes exhausting, I would not hesitate to say I would to it again, even with the knowledge of the depths of the problems I would face. It was excellent on-the-job training in supervisory skills which I will always be able to use. The experience at organization formulation and standards setting is invaluable. The close contacts and friendships with user personnel have already been valuable to me as an analyst. Most important, the experience gave me an in-depth knowledge of every admistrative computerized system at the College and their interactions. This too has been extremely valuable in my job as one of two systems analyst for the College. I also anticipate that it will help my career whether I remain at William and Mary or go elsewhere in the world of data processing.

Many of our problems existed (and still exist) because of our development over a 12 year period. Data processing centers with shorter life times or more money to spend on highly qualified personnel or consultants would not have such problems. New data processing centers have much more experienced and educated personnel to lead them and users who are more familiar with data processing. Still there exists the problem of limited funds or position justification based on student body FTE. Most new shops begin similar to William and Mary - just substitute a terminal system into a host center or a mini-computer configuration. The problem of utilization of personnel remains the same. To young small shops feeling the pressures of growth and well as to small established shops, I would urge you to seriously consider establishing a Production Control Group. For the young shop several advantages exist when the Production Control Group is added into the structure early in the history of the center:

1. Programmer idiosyncrasies are not as deeply engrained early in the center's life
2. Better utilization will be made of your highly paid personnel.

So - don't go home and hire another programmer to ease the burden of your current staff; hire a technician to take the production burden off your programmer so that he can do the job he is trained to do and for which he is paid highly.

TOWARD DEVELOPMENT OF PUBLIC TELECOMMUNICATIONS CENTERS

Timothy J. Fay
Joint Council on Educational
Telecommunications

Using examples of current policy and current practices, the author illustrates some of the many ways in which each specialized segment of the information industry depends upon several or many of the other segments of the industry. The author briefly examines some of the relevant legislation governing the telecommunications industry and calls for an integrated, cooperative approach on the part of business and government in formulating a rational plan for a comprehensive national telecommunications information and data network.

INTRODUCTION

As little as ten years ago the province of the Joint Council on Educational Telecommunications was known as "educational television". Technological, political and sociological developments during that time span have resulted in a field of endeavor called "public telecommunications". Functioning in this field today the JCET represents a consortium of leading educational and communications organizations and provides an interface for the coordination of education's interests in telecommunications. It serves as the established mechanism by which the educational community may play its rightful role in the formulation of telecommunications policy. Our perspective encompasses not only public television but CAI & CMI, communications satellites, data communications, CATV, video disc technology, fiber optic transmission systems and other public sector variants of telecommunications.

JCET's membership includes such seemingly diverse interests as the Corporation for Public Broadcasting, the American Library Association and the Association for Educational Communications and Technology, to name a few. In some way all of our members are a part of the so-called information industry. This industry is today the most significant sector of the U.S. economy. More people are now engaged in the manipulation of symbols than are engaged in the processing and handling of physical goods. The information industry employs 50% of the work force.(1)

Many of us can accept this fact with equanimity because we are actively involved in the industry. We have an instinctive if not objective feeling that we live in an information society. The transition to an information society has occurred at an almost exponential rate. In 1900 only 10% of the labor force was in the information industry. By 1960 it was 27% and by 1970 48% of the work force was employed by the information industry.(1) If we individually have adapted to this change in a stumbling, unaware sort of way our public policy certainly has not yet quite adapted to it.

In this paper I hope to convey to you a sense of the intricate interdependence of several of the many facets of the information industry and to illustrate this with examples. This paper will also provide a cursory examination of some of the relevant public policies governing this sector of the economy. Finally, I will outline a logical result of an informed and responsive public policy: the public telecommunications centers. The idea is not original, and I am not suggesting it as the only logical extension of a sound public policy. It is, however, a convenient concept for facilitating a perspective on some of the many options we must soon face in this arena as practicing professionals.

EXAMPLES OF MODERN INFORMATION MEDIA AND SYSTEMS

Electronic Mail

One study shows that 45% of first class mail originates in a computer, such as bills, and 20% of it ends up in a computer. With rising labor costs and an increasing premium on speed of transactions, it is becoming impractical to print computer data out on paper, physically transport it to a distant location, and re-encode it into a computer. (1)

Last spring a communications consultant reported at a communications law workshop that the U.S. Postal Service stood to lose 40% of its annual revenue to a de facto electronic mail industry. He reported there are 100,000 teletypewriters, 15,000 communicating word processors and one million computer terminals in day-to-day operation.(2) One author projected that by the mid-1980's the USPS will not be able to physically handle the tremendous volume of mail we will be generating in this country.(3)

The USPS surprised many observers by announcing that it is experimenting with an electronic mail system dubbed Electronic Message Service System (EMSS). The Postal Service has, in fact, already invested $10 million in an experimental system and expects limited commercial operation by 1981. Their EMSS will include such exotic components as coin operated facsimile transmitting facilities in shopping malls. They will also include facilities for direct keyboard entry of messages.(4)

Electronic Libraries

In the city of Casper, Wyoming, the Natrona County Public Library has been using two broadband channels on the local cable television system for enhancing its service to its patrons. One channel is used to broadcast instructional programs produced by the library itself. The other channel is used as an electronic reference service. A viewer has merely to call the library with a request for information and shortly the information appears on the viewer's screen at home. (5)

Many libraries are experimenting with computerized cataloging systems, intra-library sharing of digital databases via communications lines, and even computerized charging out of books. A private company in Cincinnati has introduced a complete electronic book circulation and control system which utilizes bar coded data on the books similar to the bar codes you see on most of the supermarket goods you buy. A hand held scanner reads the bar codes, and the computer decides when the book is due back, or even if it may be charged out at all, accepts input from a similarly bar coded library card and determines if the user has any overdue books.(6)

The Library of Congress is already maintaining a computerized datafile on many of its book acquisitions. The MARC magnetic tapes produced by the Library of Congress form much of the input for operational shared cataloging systems around the country.(7)

Electronic Information Services in Broadcasting

KCBS radio in San Francisco has computerized their news operation in cooperation with CBS Radio and United Press International. Using video terminals, reporters may elect to review and/or edit any stored stories, enter original stories, and later recall the stories, in proper sequence, for reading directly on the air.

In Britain you may have heard of three computerized information services utilizing the home TV set. They are Ceefax, Oracle and Viewdata. Ceefax and Oracle are two "no cost" passive systems operated by the BBC and the IBA, respectively. In these two services digitally encoded data is "piggybacked" onto unused scanning lines of the regular BBC and ITA TV broadcasts. A converter on the home TV set can be switched to display this data. News, sports, weather, stock market quotes and other data are available on hundreds of data pages. These pages are continually re-broadcast in cyclic fashion: when the viewer selects a given page for viewing, the converter "waits" for the desired page to be re-broadcast, stores it in its memory, and displays it on the screen for as long as the viewer wishes.(8)

The third service, Viewdata, is operated by the British Post Office and uses the same converter and signal standards as Ceefax and Oracle, but the data are not broadcast over the air. Instead, the user's telephone is used to convey the information to the TV converter. This system is interactive and includes provisions for adding message storing and forwarding capabilities among users. It can boast a much larger database than Ceefax and Oracle.

These systems are similar to the technology being explored in the U.S. for encoding captions on standard TV signals. Thus, with the aid of a converter, the hearing impaired viewer can enjoy not only the picture but the sound as well.

The Computer as Educator

Many of us who have been involved in the computer-based education industry have seen it grow from relatively modest promise only a few years ago to a very active industry today. The PLATO system is perhaps among the better known CAI systems and utilizes a large powerful central timesharing computer in conjunction with extensive disc storage and a data communications network. There are perhaps 2000 PLATO graphic display terminals in regular use in business, government and education. The TICCIT CAI system is another well known project and utilizes on-site minicomputers and specially modified color TV sets as graphics display terminals.

Illinois, Minnesota, Texas and California are among the states sporting statewide educational computing networks. The uses of these networks are illustrated in the Gulf Texas Multi-Regional Processing Center which is located in Houston. This center became operational in 1972 and today serves over 700 public schools in the state encompassing 700,000 students. It boasts 200 terminals devoted to administrative functions and 200 terminals are also devoted to vocation-counseling, computer-assisted instruction and computer-managed instruction.(9)

Computers, Communications & Public Services

Most of you are probably familiar with the "911" emergency telephone number which has been promoted in many areas. The idea, according to the promotion, is that 911 be the universal emergency number to be dialled for any emergency, be it fire, robbery or medical emergency. This program is now ten years old and approximately 25% of the country is served by such a system.(10) Some interesting aspects of the service include the fact that the local community or municipality provides the staffing and determines the exact configuration of the service in the area in a cooperative effort with the phone company.(11) Some of the problems worth considering include the fact that not all of the local telephone exchanges are equipped with the necessary facilities for delivering a centralized "911" service. Another reason for the slow growth of the service on a national basis is that the phone company exchanges frequently do not coincide with the local political boundaries.

The Japanese are experimenting with a broadband information network for use by the public to augment their existing public broadcasting service and includes a facsimile newspaper, an information retrieval system with TV display and a two-way multiple-choice citizen response mechanism for citizen participation in televised public meetings.(12)

In Reading, Pennsylvania, a local non-profit corporation operates a two-way broadband interactive system for providing social services to senior citizens. The operating corporation represents a broad range of public and private institutions in the community. The system consists of three neighborhood telecommunications centers which have TV cameras, monitors and two-way communications gear interfacing with the broadband network and the standard phone network. Daily sessions are broadcast dealing with public services for senior citizens, including "live" meetings using the two-way capability for input and response from the viewers. (13)

ISSUES THAT NEED TO BE ADDRESSED

Before laying down a rough blueprint for a public telecommunications center it will be useful to review some questions which need to be kept in mind. The scope of this paper is too modest to formulate possible answers to these questions, but raising them will serve to help focus attention on the issues which will be raised by the subsequent proposal. For instance, who should foot the bill for these fabulous information services? Should they be supported entirely by public money? An underlying assumption of our democratic society is that the public needs unrestricted access to information so it may make informed decisions in the democratic process.(14) The Library Services Act of 1956 extended federal aid for the purpose of making library services available in rural areas. The Rural Telephone Act charged the REA with the authority to use federal funds to ensure access to our ubiquitous phone network among the nation's rural communities.(5) The premise was simple: citizens in the cities are not the only ones entitled to information. The objective was to achieve greater parity of service between urban and rural areas and possibly to achieve or at least encourage a more even population distribution.

The Communications Act of 1934 charged the Federal Communications Commission with the responsiblity of ensuring the best possible communications system for our nation at a reasonable cost. The key phrases are "best possible" and "at reasonable cost". The history of regulation of the rates of the common carriers shows that we have yet to satisfactorily answer the question of "reasonable cost". Another prime example of the concept of equitable distribution of communications services at reasonable cost predates modern telecommunications altogether: the postal service with certain rates uniform ensured virtually all citizens access to an inexpensive and relatively timely mechanism for maintaining business and private correspondence.

The federally-assisted public broadcasting system in this country is a notable example of a hybrid funding mechanism for public telecommunications services, albeit complex and cumbersome. The public broadcasting service receives federal funds on a matching basis against other non-federal money raised by the licensees themselves. Direct federal grants are also part of the mechanism. State education budgets frequently supply a substantial remainder of the funding requirement, with possible inputs from local or regional agencies which may, in fact, be the licensee. One would hope that the support system of a more comprehensive national public telecommunications system would be more practical to administer.

Who should spearhead the planning and development of a network of public telecommunications centers? A panelist from a major computer manufacturer seemed to echo many informed students of the field at a recent seminar on metropolitan telecommunications services sponsored by the National Research Council. In essence he said the information and telecommunications industry is so large and complex, the rate of technological advance so rapid, and the task of formulating a comprehensive and equitable plan so formidable that the federal government needs to assume the role of initiator.(10)

Finally, remember that the creators of the information, whether it be news articles, textbooks, TV programming or computer programs are entitled to compensation for their work. We subscribe to the concept of intellectual creation as property in this country. That an equitable compensation system is difficult to establish even with today's telecommunications distribution of such property is evidenced by the many years Congress spent in revising the copyright laws.(1) The question of compensation will become even more complex as our telecommunications systems evolve.

POLICY AND REGULATION: IMPLICATIONS FOR THE TELECOMMUNICATIONS SERVICE

Data communications is perhaps one of the fastest growing segments of the burgeoning information industry. The common carriers, by virtue of their regulated monopoly status, play a very key role in the success of innovation in the area of telecommunications services. Regulation and public policy represent important moderating variables in the relationship between the carriers and their data/telecommunications customers.

The telecommunications industry, like other public utilities, is one of the exceptions to the concept embodied in our antitrust laws relating to monopoly. The Communications Act of 1934 establishes the ground rules for government supervision of this so-called natural monopoly.(15)

Two recent rulings of the FCC are of particular interest. In 1976 the FCC ruled that customers of the telephone company could connect non-telephone company equipment directly to the phone network without protective devices. This permits customers a wider range of competitive equipment choices and can result in lower costs to the user. A second and perhaps more significant ruling involved "resale and sharing" carriers (formerly called "value-added" carriers) who purchase wholesale communications capacity from the common carriers. They then re-sell the capacity to a number of smaller customers thereby passing on volume savings. These resale and sharing carriers may also perform some additional service, such as storing data, modifying it in some way, and re-transmitting it over the network for the customer.

The FCC has ruled that such carriers would be regulated but not as stringently as the common carriers. Additionally, since resale and sharing carriers have no monopolistic services with which to cross-subsidize their communications services, they would be allowed to engage in data processing customer services as well as communications services. More to the point of this paper the FCC ruled that they would also regulate private data processing companies _if_ they engaged in supplying services which are principally communications services.(15)

An Aging Regulatory Framework

The Communications Act of 1934 is an antiquated piece of legislation modelled after the even older 1887 Interstate Commerce Act. It was written during a time when communication by wire or radio was viewed as similar to the physical transportation of goods.(5) The House Communications Subcommittee is currently working on an extensive revamping of this piece of legislation which hopefully will take into account the tremendous technological distance we have moved since the initial 1934 Act. Among the options being considered by the subcommittee is the establishment of a blueprint for a modern public telecommunications system utilizing existing as well as emerging technologies. The subcommittee has, in fact, been quite interested in observing the activities of the Japanese in this field, as mentioned in a previous example. Another possibility under consideration is the development of "public telecommunications centers" which could provide one- and two-way interactive services through a combination of over the air broadcasting, CATV, computer-mediated services, and others.(12) Such centers could be based upon the extant public broadcasting facilities across the country, or they could be implemented through a different extant mechanism, eg., libraries with their tremendous information resources and increasing involvement in telecommunications.

New Look at Existing Service

Ten years after the first Carnegie Commission on Public Broadcasting, a second commission, dubbed Carnegie II, has been convened. The commission will examine the past ten years' performance and will try to see what needs to be done for the future. Will the commission recommend the addition of services to the system which are non-broadcast in nature?

Future Funding Alternatives

A new bill has been drafted called the Public Broadcasting Financing Act of 1978. The bill, H.R. 11100, represents the administration's response to the needs of the public broadcasting system for additional and longer range funding and other improvements. Perhaps the most salient point in the bill relating to this paper deals with the Educational Broadcasting Facilities Program (EBFP). As currently instituted, the EBFP allows federal grants to public station licensees for construction of broadcast-related facilities. Under the proposed legislation these federal funds may be used for construction of non-broadcast facilities as may be necessary for the delivery of service. Further, the grant applicant need not be a station licensee. A library, for instance, could apply for funds. So we can see a trend toward reliance on other than the tradional technologies and institutions for telecommunications delivery of "public" programming and services.

Computers & Communications: Integration or Segregation?

The FCC is in the midst of its second Computer Inquiry. In the first such inquiry the commission drew the line between regulated common carriers and non-regulated data processors: a company engaged principally in providing communications services would be subject to regulation by the FCC while a company engaged principally in data processing (or, simply not engaged in communications) would not be regulated. Since the first inquiry, however, it has become evident that it is not so easy to make the distinction due to rapid advances in technology and marketplace demands for unique combinations of computing and communications services. The outcome of Computer Inquiry II remains to be seen.

Juggling Policy, Technology and the Status Quo

The regulation of the telecommunications segment of the information industry has been convoluted at best. In addition to the FCC, there are over 40 government entities involved in telecommunications policy. A few such include DOD, NASA, HEW and NSF.(12) This is another example of the "patchwork syndrome" we saw manifested in the funding process for public broadcasting. Emerging policy will hopefully reflect an awareness of the need for greater coordination in policy planning regarding such a multi-faceted industry whose segments are so heavily interdependent.

Be aware also that the history of regulation shows a somewhat protectionist attitude toward the industry insofar as having a sensitivity to the problem that could be caused by too rapid change in policy. Technological advances can virtually obsolete billions of dollars of plant and equipment investment if unchecked by policy. The phone company, for instance, plans some of its $70 billion investment for as long as 40 years.(1) There are still 9,000 hand crank telephones in regular use in rural America. 90% of rural America is still served by electromechanical step-by-step switching systems which were common in urban areas 30 years ago. The message is clear: dramatic short- or medium-term changes in the telecommunications industry on a nationwide scale are not likely.

On a more positive note, or at least more "change oriented", the FCC has shown a tendency to foster competition among the communications carriers, much to the consternation of AT&T. The ruling on resale and sharing carriers and the interconnection of terminal equipment are examples of this tendency. Specialized private carriers such as MCI are helping to spur AT&T to provide competitive services which might not have been provided if AT&T had only been considering their 30 and 40 year depreciation schedules.

TOWARD A PUBLIC TELECOMMUNICATIONS CENTER

The federal government has shown a willingness to fund experimental projects and to otherwise contribute "seed money" to the development of improved means of delivering public services utilizing modern technology. As cited earlier it has been the national policy for some years to improve rural telecommunications services, both in the interest of equity and to encourage a better population balance between rural and urban areas. Thus the REA has been engaged in experimental programs with broadband communications lines providing a variety of public services and even providing common telephone service via the broadband cable. One of the many challenges facing the public brodcasting system today is how to improve coverage of the rural areas given present funding constraints.

The technology certainly is in place to allow vast improvements in existing telecommunications services and to permit radical new services to be added. The shaping of public policy to encourage these developments would seem to have begun.

What's a Public Telecommunications Center?

Let us assume that the base for the establishment of the Public Telecommunications Center (PTC) will be the extant public broadcasting outlets across the country. A second major assumption is that the services offered by the PTC would be consistent for the most part with official or de facto policy and regulation governing telecommunications in the public sector. A third and very important assumption is that the requisite broadband (CATV-type) network which connects private homes, businesses and institutions will not be operated by the PTC. The CATV industry, with its flexible broadband technology, would be a likely carrier. An advantage of CATV handling the interconnection would be that it, like the PTC, is a locally based phenomenon. In theory such an arrangement helps ensure more responsiveness to local needs. CATV systems are licensed to municipalities and are required to provide certain minimal provisions for public services such as local government access and citizen access to the broadband network.

An illustrative if not exhaustive list of the services which would logically be offered by PTC's follows.

- o Cultural and educational TV & radio programs (as currently offered by public broadcasters)
- o Instructional and training programs
    - * In-service training
    - * Interactive "live" instructional programming (viewer participation via two-way response facilities)
- o Library services
    - * Bibliographic search
    - * Information recall & display Facsimile of data
- o Various computerized services for the home
    - * Financial management & budget programs
    - * CAI & CMI alone or in conjunction with the library service
- o Pulbic services
    - * Security monitoring of homes & businesses
    - * Remote health care, routine and emergency
    - * Social services for the disadvantaged
- o Electronic mail and facsimile

Just as the public broadcasting system in this country is an alternative to, and not a replacement of, the commercial broadcasting system, so too the PTC would (more or less) peacefully coexist with private telecommunications services. Examples of the services that commercial interests might provide on the broadband interconnecting network follow.

- o Commercial TV & radio programming
- o Shopping in the home
- o Magazine- and newspaper-type services
    - * Electronic video information display
    - * Hard copy capabilities
    - * Indexed search capabilities
- o Electronic banking and other financial services
- o Telephone service
- o Computer games

With technology changing so rapidly it should be noted that in the interconnecting network, by the time the PTC is implemented, could be of a very different nature than a coaxial cable system. Ultra-high capacity optic fibers hold a good deal of promise as an alternative. Also, advances in satellite technology could result in rooftop antennas handling the two-way broadband connection.

Finally, if a network of PTC's were to develop it is likely it would happen in a multi-staged fashion. For instance, a conceivable first stage might be the addition of Ceefax-like data transmission via the public broadcasters with provision for phone interaction. Another stage could include the introduction of a broadband connection to the home, either followed or preceded by introduction of home-user oriented computer services via inexpensive terminals. Still another stage could be the integration of the by then in place electronic mail service into the PTC.

Our Role in Achieving a Comprehensive National Public Telecommunications and Information System

It is important that each of us maintain our proper professional perspective on the information industry. The systems programmers for the statewide educational computing networks, for example, are no more the axis around which the information industry orbits than are the school administrators participating in the program. Librarians have no more of a monopoly on information services than operators of the data communications networks. And increasingly the common carriers are no longer the center of the data communications universe; private land- and satellite-based data networks are now offering competition. In short, we are indeed intricately dependent on many of the different disciplines within the information industry quite outside our own individual professional niches. Our self-interest in pursuing our own goals must be enlightened; our input to the policy making bodies must reflect our concern for the overall development of these new and essential services in both the public and private sectors.

It is nonetheless difficult to continue to devote the necessary energy to our professions while remaining abreast of important developments in the tumultuous sea of current policy making activities. To remain informed on every legislative and regulatory development which will have an impact on you and I in our fields would be more than a full time job. It is therefore important that we encourage interdisciplinary associations and other arrangements as appropriate which purpose will be to (a) act as information resource regarding current policy developments; (b) act as a focus for our informed input to the critical policy making process; (c) foster development of interdisciplinary solutions to societal needs. In approaching the idealistic goal of "the best possible service at reasonable cost" we can ill afford a feudalistic attitude toward segments of the industry other than our own.

I personally accept as an article of faith that we can and will see the implementation of much needed modern telecommunications services, both public and private. From my perspective with the Joint Council on Educational Telecommunications I see that we are in the midst of an exciting and challenging phase in the development of public policy regarding information and telecommunications services. The implications for society at large are vast even if the details are indeterminate. Let us adopt a forward looking stance and embrace the positive and constructive aspect of the rapid changes which threaten to overwhelm us, and let us discard the counterproductive aspects of change as the informed and responsible professionals we are.

REFERENCES

1. Ithiel de Sola Pool, "Policy Choices for the Information Age", in Refocusing Government Communications Policy, Aspen Institute, February, 1976.
2. "Electronic Mail Could Cut Deeply in Postal Revenues", Communications News, March, 1977.
3. Ralph Lee Smith, The Wired Nation, Harper Colophon Books, 1972.
4. Data Communications, December, 1977.
5. American Library Association Statement to the Communications Subcommittee on Revision of the Communications Act of 1934, November, 1977.
6. "Electronic Book Control for Central, Remote Data", Communications News, January, 1978.
7. Aronofsky & Korfhage, "Telecommunication in Library Networks: A Five Year Projection", Journal of Library Automation, March, 1977.
8. The Globe and Mail, Tuesday, July 12, 1977.
9. Communications News, December, 1977.
10. Seminar on Potential of Telecommunications for Metropolitan Areas, National Research Council, February 28, 1978.
11. Communications News, January, 1978.
12. Options Papers, staff papers for use by the House Subcommittee on Communications, May, 1977.
13. Mitchell Moss, "Interactive Cable Television Research in the U.S.", Exposition Proceedings, First International Telecommunication Exposition, October, 1977.
14. Virginia Mathews, Libraries for Today and Tomorrow, Doubleday & Company, 1976.
15. Philip Nyborg, "Computer Technology and U.S. Communications Law", Telecommunications Policy, December, 1977.
16. Office of Telecommunications Policy, Interagency Committee Report on Rural Communications, December, 1977.

THE IMPACT OF COMPUTER-BASED INFORMATION SYSTEMS
UPON SCHOOL AND SCHOOL DISTRICT ADMINISTRATION

Tom Hansen and Dan Klassen
Minnesota Educational Computing Consortium

Jim Lindsay
Total Information for Educational Systems

This paper is a summary of a two-year study that has been supported wholly by the National Institute of Education. The study was conducted by the Special Projects Division of the Minnesota Educational Computing Consortium (MECC). The primary objective of the study was to provide new information on the nature of administrative computer use in elementary and secondary schools. We have investigated the nature of administrative computer use and impact as it occurs at three levels of management activities: (1) Strategic Planning, (2) Management Control, and (3) Operational Control. In-depth interviews were conducted with 134 employees in eleven Minnesota school districts. The districts chosen for the study are all members of the TIES (Total Information for Educational Systems) cooperative which has been in existence since 1968.

Background

The operation and management of schools are increasingly complex tasks. Given the problems and concerns spawned by rising costs and dwindling financial resources, educational decision-makers are constantly seeking ways to help them keep schools operating in a manner consistent with the goals and expectations set for them, and within the financial resources available.

In order to successfully manage educational institutions, personnel and programs, many school administrators have begun to employ computer technology and computer-based information systems. The purchase and development of computer-based information systems designed specifically for use in the administration of schools and districts, according to a recent nationwide study, is occurring with increasing frequency. In fact, given current growth rates, computers will be used by every school district in the country within the next decade.

The adoption and use of computer-based information systems is based on the expectation that these systems will produce favorable results: they will help save time and money, they will increase flexibility and responsiveness, they will produce more accurate, detailed and accessible data, and so forth. Unfortunately, few attempts have been made to identify the nature and scope of the use of computer-based information systems or to examine their impact on the operation of schools.

In an effort to shed light on the questions surrounding the use and impact of computer-based information systems, the Special Projects Division of the Minnesota Educational Computing Consortium with financial support from the National Institute of Education, undertook a field-based study that focused on a computer-based information system that can process and produce information dealing with students, curriculum, personnel, facilities/equipment/supplies, and finances. The system chosen for the study is made available through TIES (Total Information for Educational Systems), a cooperative developed and operated by fifty school districts in Minnesota. The TIES-member districts were selected for analysis because of their relatively long and stable history in using computer services, and because of the representativeness of the services offered. In-depth interviews were held with 134 school district employees in eleven of the fifty districts. Persons interviewed included superintendents, assistant superintendents, business managers, accountants, personnel managers, administrative assistants, directors of research and evaluation, principals, assistant principals, counselors, teachers, clerical and secretarial personnel, and each district's information system coordinator.

These district employees perform a wide variety of functions. These functions may be categorized as dealing primarily with strategic planning, management control, and operational control.[1] Strategic planning, as the name implies, includes long-range planning and policy-making. For example, decisions about future facilities would fit into this category. Management control involves the development, supervision, and evaluation of programs/activities that are designed to meet the purposes of the policies and long-range plans. Examples would be the development and analysis of alternative employee salary schedules, or the evaluation of the teaching staff. Operational control refers to specific and often routine tasks such as the development of mailing lists and labels, and payroll processing.

This study investigates the ways in which the computer-based information system interacts with the various functions performed in a school district.

Findings and Conclusions

Availability

The computer-based information system used in the school districts studied includes an integrated collection of computer-based administrative applications (e.g., scheduling, mark reporting, financial forecasting). Most of the applications were specifically designed to perform operational control tasks. The availability of a rather large number of computer-based applications designed to perform operational control tasks is similar to most other educational, as well as non-educational organizations. Since computers can perform clerical and computational tasks very efficiently, and since the development of these applications is quite straightforward, it is not surprising that this type of application outnumbers applications designed to support management control and strategic planning. However, after several years of experience in using computer technology and computer-based information systems some users are beginning to reach a higher level of sophistication and they are actively seeking ways to enhance current applications and develop new applications that will facilitate and support strategic planning.

Since the districts studied shared a common computer-based information system, the applications that are available, for the most part, reflect the desires and needs of the majority of the users. The cooperative nature of the service does, however, mean that the individual district must learn to use the available system even if it is not exactly what they want or need, given their unique problems and structure. If a district does not have an individual(s) assigned to facilitate the use of the applications who knows enough about them to be able to adapt them to the district's special conditions, the availability of the computer-based information system for that district remains primarily at the clerical/computational level. However, if the district's information system coordinator understands the complexities of the system file structure, can use higher level information linking capabilities of the system, can employ projection applications, and makes use of the research and technical consultation services, the availability of the system as an aid serving management control and strategic planning is expanded considerably. As a result, the actual availability of applications to the various users in a district is affected by both the collective decision of the cooperating districts as to what applications and services should be offered, and the ability of the individual(s) charged with delivering and supporting the specific applications within the local district. The actual availability (the degree to which the district can utilize the system in support of all three levels of operation) can be effectively and significantly increased if the district utilizes a person(s) who has a good working knowledge of computer technology and related information systems, and provides that person(s) with sufficient resources and authority within the district.

Since all of the cooperating districts pay for computer services based on the same payment schedule, cost is not a significant factor impacting the availability of the basic applications. All districts, in essence, have the same applications available to them. However, because most smaller districts as well as some larger ones are faced with financial constraints which influence their ability to employ a person with the expertise and time needed to adapt the system to district needs, cost does become a factor determining actual availability.

Use

As might be expected, considering the nature of what is available, the greatest use of the computer-based applications is at the operational control level; this involves, for example, the production of class lists, mailing labels, and attendance reports. Functions of this type tend to be performed on a regular and reoccurring basis, so the frequency of use of the applications designed to perform these tasks is quite high. Use of the computer-based information system decreases as one moves toward the strategic planning level of district operation.*

---

*For more examples, refer to the appendix.

There are several basic reasons offered for the less frequent use of the system at the management control and strategic planning levels. First, and this is particularly true at the strategic planning level, the major decisions made and the tasks performed are long-range and thus naturally occur with less frequency. Next, the tasks to be performed and the decisions to be made must incorporate many factors other than just computer-stored and summarized information. Since it is difficult, if not impossible, to know exactly how a decision-maker actually makes decisions or to determine what factors influence a decision and to what extent, the importance of computer-produced information is difficult to determine. While numerous examples were cited by the decision-makers interviewed (for example, the importance of census data in reaching vital decisions as to future staffing needs), the relative weight given the information by the decision-maker remains unknown. Experience, judgment, and understanding of the situation are all factors which must be entered into the decision-making formula. Third, since many decisions made at the top management level are influenced by factors which are beyond the control of the school administrator, the important information for decision-making is generated by the environment external to the school. For example, a school district's future funding and facilities needs are directly influenced by population growth and migration trends. Except for enrollment and revenue projects, there is very little information within the computer-based information system that can be used to analyze external factors; to analyze community attitudes and trends, to make comparisons and contrasts with other organizations, or to examine the impact of changing legal and structural trends and requirements. Finally, it remains true that there are many long-range, strategic decisions made that are based on information concerning the district's internal operation and situation. Decision-makers want and need summary information, reports of exceptions to normal operation, and internal trend information. Currently, this type of information is usually developed and compiled manually at a lower operational level within the district, although some information system coordinators have produced such information through the use of the system enhancements that are available. Indications are that such information is valuable to decision-makers and that those who have received it are demanding an increased capability of the computer-based system to produce it more easily than can be done using current procedures. This type of demand seems to offer the greatest potential for the expanded use of the information system at the strategic and long-range planning level.

Impact

It is evident that the available computer-based applications are used to perform a wide variety of important tasks in the school districts. As a consequence, the impact of computer use is found throughout each district, although not to the same extent in every district. While some persons report a desire to change or enhance the current applications, few persons feel that the computer and the information system it supports are producing unfavorable, or negative impacts. On the contrary, most view the computer as a tool that is enabling them to perform many vital educational and administrative tasks; a tool that they would not want to do without. The computer, according to most users, allows them to perform many tasks and to accomplish many objectives that would not be attempted without the aid of a computer. Listed below, in summary form, are some of the major impacts identified during the study. Except for the instances noted in numbers 5 and 6, the impacts were perceived by those interviewed as being favorable.

1. For many people, a significant amount of time has been freed from clerical tasks. While for some, this "freed" time is absorbed by computer-related duties, for others the time is truly available to perform other tasks.

2. More accurate personnel, student, and financial records can be kept. Records can be accessed more easily and the stored data can be sorted in a multitude of ways, thus facilitating flexible reporting to school personnel and external agencies.

3. More accurate projections of student enrollments can be produced, thus greatly aiding in planning for facilities and personnel needs, and in communication with constituents.

4. Decentralization of budgetary development and control is made more feasible because personnel receive accurate, detailed, and relevant budget information. Many more line items can be included in the chart of accounts.

5. More frequent and flexible scheduling of classes can be performed. More electives, and more classes of varying lengths can be offered. In schools with less complicated traditional schedules, this benefit is not

realized, and the impact of computer use in scheduling is perceived as unfavorable by a few persons.

6. In most schools, a longer period of time now elapses between the end of a term and the distribution of student progress reports. Some teachers and administrators view this as an unfavorable impact of computer use.

7. Districts can more easily conduct research studies of their activities, such as student follow-up studies that aid curricular decisions.

8. Significant cost savings can be realized through use of the computer to aid in developing bus routes. Although the computer program does not automatically determine routes, the data stored and processed by the computer greatly aids the building of efficient routes and the performance of day-to-day transportation tasks.

As this summary list illustrates, significant favorable impacts were found to exist across a wide range of functions and tasks performed in the operation and management of school districts. In the districts which make the most extensive use of the computer-based information system nearly all of these impacts, and many other significant favorable impacts, were found to exist.

Most of the impacts mentioned by those interviewed are related to the operational and management control levels of district operation. At the strategic planning level the most significant impact seems to be that accurate, timely and comprehensive census (population) data are enabling administrators and school board members to better anticipate changes in enrollments. Since for many districts the decline in enrollment is the root of most of the major problems and concerns facing the district, good census data are very valuable to top decision-makers. Because financial aid is linked directly to enrollments, and because operating costs do not necessarily decrease in proportion to enrollment decline, it is paramount that declines be predicted early and accurately.

Computer-stored and produced budget information is also important to the top administrators as they seek ways to use scarce resources more efficiently. In districts that are not facing the problem of declining (or shifting) school age populations, the perceived need for accurate census data and related strategic planning information is less evident, and the impact of the computer in decision-making at the strategic level seems minimal.

Value of Computer Use

Two different levels of benefits related to computer use have emerged in this research study. First, there are those fairly observable and tangible impacts of computer use which, in the opinion of the people who were interviewed, are beneficial. For example: timesavings; more accurate and timely data; the capability to schedule classes more flexibly; the capability to decentralize budgetary development and control, and easier reporting to external agencies.

Computer use enables the district to realize these benefits. Most of these benefits would not be realized or, at least, would be more difficult to achieve if a computer were not used to process and produce information.

While the evidence generated by this study supports the conclusion that these types of benefits can be obtained, the question of their ultimate value remains. Is having the ability to obtain these benefits of any real or lasting value to a district?

Most interviewees suggested that there is another level of benefits that derive from and are related to the first. Better decision-making, better long-range planning, more effective resource management, more time to work with people, and better educated students are claims often heard. For instance, in districts that have a decentralized budgetary development and control process, the claim is made that by placing budget responsibility and authority with the persons who are closest to the student, the curriculum and the learning resources, more effective planning and management of resources will follow. The computer, it is argued, enables the district to provide personnel with the type of information that is necessary before they can be asked to be accountable to management for their specific area of responsibility. It is not the purpose of this study to determine whether or not this redirection of accountability is of ultimate value to the district. It is fair to conclude, however, that the computer at least offers a district the potential to make changes of this type that are thought to be important.

While the computer-based information system is a valuable tool, one should not expect computer use to automatically produce an improvement in the quality of management in the district. In school districts where careful planning is recognized as being essential for effective management, the computer becomes a valuable part of that planning process, and better planning may result. However, computer availability and use does not of itself lead to careful planning, nor does it create the desire to engage in long-range planning. It is not a cure for poor management; but it can be of immense value to the person looking for ways to manage more effectively.

The computer is one intervening variable in the complex process of educating students. It is a tool that can help administrators, for example, to bring students and teachers together more efficiently so that learning may take place. However, no evidence was found that "better learning" results from district use of a computer to aid administrative functions.

Nevertheless, the pervasiveness of the more observable and tangible benefits, and the potential for realizing the more tangible benefits, lead us to conclude that the investment of time and money yields very worthwhile returns. In the few districts that have gone beyond use of the standard applications, and have used special enhancements to meet unique needs of management, the value has been even greater. They have effectively expanded the basic computer-based information system into a computer-based management information system. The expanded system has increased value because it meets the information requirements of a broader spectrum of district personnel.

It is important that potential users of computer-based information systems understand what such systems can and cannot do for their districts. If they "buy into" such systems solely because they expect to reap the more intangible benefits, they may have difficulty in the future justifying the investment. However, if they attempt to justify the investment on the basis of the more tangible and observable benefits that are to be expected, evidence to support the wisdom of the investment should be readily apparent to both users and observers.

Notes

[1] Anthony, Robert N. Planning and Control Systems: A Framework for Analysis. Boston: Harvard University Press, 1965; Hentschke, Guilbert C. Management Operations in Education. Berkeley, California: McCutchan Publishing Corporation, 1975, pp. 433-442.

APPENDIX

EXAMPLES OF USE

	STUDENTS	CURRICULUM	PERSONNEL	FACILITIES/EQUIPMENT/ SUPPLIES	FINANCE
STRATEGIC PLANNING	- Enrollment Projection	- Use of follow-up studies of graduates to aid curriculum content decisions. - Studies of curricular goals of community and staff.	- Census data used to plan future staff needs. - Seniority lists.	- Census information for projecting school needs.	- Census information incorporated into finance projections. - Expenditure and Revenue Forecasting.
MANAGEMENT CONTROL	- Statistics on students/families (summarized). - Enrollment Projection --school level. - Mark distribution analysis (looking at students). - Statistics on male/female sports participation.	- Mark distribution analysis (looking at subjects). - Assistance with evaluation of basic skills programs.	- Salary schedule simulations for contract negotiations. - Mark distribution analysis (looking at teachers). - Information for staffing changes. - Determining staff development needs.	- Census data for school closing and construction decisions, and boundary changes. - Equipment and supplies allocations based on enrollment projections.	- Enrollment projection used to decide on resource allocation. - Budgetary development and control. - Exception noting--90 percent spent.
OPERATIONAL CONTROL	- Mailing lists and labels. - Statistics for reports. - Record keeping. - Honor point lists. - Report cards. - Incompletes lists. - Failures lists. - Rank ordering. - Annual attendance report.	- Class lists, student programs. - Scheduling students into classes. - Aid in building master schedule (conflict matrix). - Teacher prepared test scoring. - Inventory of media resources, sorted by subject, grade level, etc.	- Personnel records. - Leave lists. - Seniority reports. - Information for state and federal reports. - Employee directories.	- Locker lists.	- Monthly financial reporting. - Payroll processing.

ON-LINE REGISTRATION AT EAST TENNESSEE STATE UNIVERSITY - JUST A FIRST STEP TOWARD AN INTEGRATED DATABASE

Stanford H. Johnson, Director
Computer Services
East Tennessee State University
Johnson City, TN 37601

On-line "interactive" student registration at East Tennessee State University saves time, eliminates confusion, reduces disappointment and frustrations for students, relieves faculty of clerical detail, smoothes the workload for clerical workers, and provides administrators with accurate up-to-the-minute information. Yet, continuous registration is still regarded as just one step toward the real goal--an on-line management information system supported by a truly integrated database.

East Tennessee State University (ETSU) is located in the northeast corner of Tennessee surrounded by some one million people within a fifty mile radius. The main campus in Johnson City encompasses 362 acres with an extention facility of 95 acres in Kingsport, Tennessee. ETSU's enrollment is about 10,500 where more than 1,400 of these are graduate students. The university is composed of seven colleges and schools. There are over fifty departments, three off-campus centers and some 2,000 courses are offered in support of 60 undergraduate and 30 graduate programs. There is a faculty of about 500.

The present computer system, a distributed network, is used in support of the academic, administrative and public service functions of the institution. Academic services include both instruction and research. Instruction comes in two varieties; first, the study of the computer as a subject (the computer science program), and secondly, the use of the computer as a tool of instruction. To some the computer is a subject of research but its primary use in this regard is as a tool of research.

Administrative services are those activities related to student services, institutional support, operation and maintenance of plant, scholarship and fellowship, auxiliary enterprises and library.

Current administrative services and projects are designed to aid university officials in their day-to-day operation. Provided is a telecommunication system with on-line data which is capable of direct inquiry and on-line update as normal office transactions occur. Data captured from these transactions are assembled into logical subsets, thereby providing the database necessary to the effective and efficient management of the university.

These data are viewed as belonging to one of five subsets. These subsets are: student, personnel, course and schedule, finance, and facilities. Student data are classified into four areas: prospective, applicants, enrolled and historical. Current services provide on-line application entry, registration, schedule build and update, drop and add, housing assignments, financial aids status, fee assessment, billing and fee collection, and instructional workload update. Many batch services (such as reports, mailing labels, and day-to-day operating information) are provided.

Present and planned projects are directed toward the extension of services for prospective students and improving the application, enrollment and reporting process. Only summary type historical data is now on-line; thus, a major project under consideration is the placement of detailed student course history on-line.

An on-line financial accounting system, a budget preparation system as well as a personnel payroll and faculty activities analysis system are in their final stage of development. Some parts are now being implemented. Live test runs were made on the faculty workload update this fall and will be fully implemented Fall 1978.

The remaining projects under development necessary to completion of the on-line management information system, is facilities data. Also underway is the search for a battery of management tools such as student flow models, course load matrices, cost simulation models and resource prediction models. All of these tools are necessary to effective and efficient management and must be compatible with the integrated database management system.

As we are primarily concerned with student registration in the paper, it is important to understand its place in the total system. In fact, the data that reflects the outcomes of the registration process is exactly what this game of managing an educational institution is all about. Four times a year, under the quarter system, we match student course needs with the institution's resources. If at this point we fail to capture the data necessary to measure the outcomes of this activity, the game is lost. Directing students to class and producing a class list are not the main objectives of the registration process, although they are important ones. The game is to capture the data necessary to model the institution's behavior and thereby make those management decisions sufficient to effective and efficient educational practices.

The schedule then becomes the centerpiece of the database. (See Figure 1) It is here we capture the outcomes data and coordinate it with data in other areas of the database to provide the required management information.

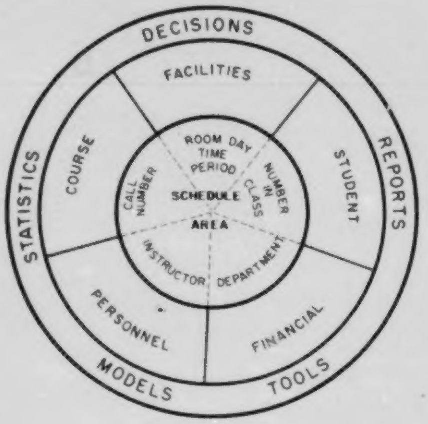


Figure 1

MANAGEMENT INFORMATION
DATABASE

Let's look at some details related to ETSU's on-line interactive registration system.

The most important ingredient in this refinement process, known as education, is the student. Although this refinement takes place throughout all institutional activities, a large portion is believed to occur in those activities related to instruction and research. Thus, registration is the process through which we schedule these activities and make record of the resources utilized within each step in the process.

Upon making application and being accepted, the student's demographics and previous educational data are entered on-line through CRT terminal devices. Each new and returning student is informed of the registration period and the appointment schedule. The latter is simply a function of the hours earned toward a degree program.

Students may not register prior to their appointment period but may do so any time thereafter. Thus, the data for all students eligible to enroll at the beginning of the registration period are on-line and each additional applicant is placed on-line as the transaction occurs. It is possible for a student to appear on campus, acquire an application, be accepted, obtain his class schedule, pay his fees within a few hours and return to campus next on the first day of class.

Examination of the appointment period timeframe will reveal one important factor contributing to ETSU's success in on-line registration. That is, the process has been spread over a long period of time. We call it a continuous registration system. This practice enables us to register 10,000 plus students on five CRT's at the scheduling point; three CRT's at the fee collection station; admissions, records and the veterans office use one each; housing and financial aids share one. Two devices are also used by the Office of Continuing Education as schedule points, one is remotely located at the Kingsport Center and is used also for fee collection. The other is on campus and is used primarily to enter materials related to the other off campus centers. Thus, a total of fourteen CRT's and two printers are utilized in this process. However, the number and location of terminal devices are not rigid. They are moved to the area of greatest need for service to the student.

Upon obtaining a schedule of classes and a course request card, the student will determine a course selection. Advisors are available upon request and are required only for freshmen and first term transfer students. A different colored card is used to indicate that an advisor signature is required. At ETSU learning to decide one's own destiny is considered to be a part of the educational process. However, if the student and/or his advisor have need for immediate accessibility of data regarding course availability, they have it via the terminal and the operator.

Before finalizing the course selection and before attempting to register, the student should check one of several lists showing classes which are 80% or more filled. These lists are posted each morning reflecting the enrollment as per the end of the previous day. However, the same lists for any percent occupancy are available via the CRT's reflecting up-to-the-minute enrollment. Similar lists showing enrollment data for all classes are made available at least once per week to college deans and departmental chairpersons and are produced daily as the first day of class approaches.

Besides listing the call numbers of the selected classes on the course request card the student must also indicate those classes being repeated, those being taken for graduate credit and/or those being audited. Decision on graduate credit apply only to senior level undergraduate courses. When course history is brought on-line the repeat decision will be unnecessary and all prerequisites for each selection can be checked. The student is also requested to list four alternate courses that may be substituted should a problem arise with the initial selection.

Once a student fills out his card and makes a copy for his own record, he gets to a terminal operator within a couple of minutes. Assuming the card is properly filled out with accurate, nonconflicting course numbers, as nearly all are, the average time to complete a student request at the terminal is thirty-four to forty seconds. However, some lines do occur as the first day of class approaches. We have noticed this to be particularly true at the beginning of the winter quarter. January 3, 1978, we serviced 1,007 students over a thirteen hour period. This is the greatest number of students yet, to register in one day. Also, this is the worst case for problems due to full classes and the general tendencies of students who procrastinate to be error proned. These students were serviced at five terminals for an average of 3.87 minutes per student.

Now there appears to be a considerable discrepancy between these two averages. Let me explain. The thirty-four to forty second average is for those trouble-free selections. For those with problems, let's look at the kinds of services that were rendered within the four minutes. A class is full--then we look at all sections of that course to determine which is still open and the time it is offered. To prevent time conflicts upon making a substitution, other courses must be changed; thus, they must also be checked as to the number of seats available and the time of offering. The student may also have an authorization from the departmental chairperson to increase the number of seats available. This is done on the CRT at this point. Although these problems destroy the overall throughput average, it is here that the interactive process pays off for all concerned. Without such interaction the student might find it necessary to visit several different offices for decisions from people who are not knowledgeable of up-to-the-minute data. This would require several hours, many steps and all the other frustrations that accompany this kind of problem.

The number of students participating in last minute registration is on the increase. Our rate of increase is reflected as follows: 781 registered January 2, 1976, 993 registered January 2, 1977 and the above 1,007 registered this January. The explanation seems to lie in the complacency acquired by those students who have never experienced a card distribution center. Since we initiated this process in 1975, only our seniors have had this experience. Registration is easy, no problem; thus, less planning is exercised by the student and last minute registration occurs. Of course, there are other factors which contribute but complacency appears to be the chief among them. We fully expect 1,100 next January and they will complain about the lines and feel no responsibility for their existence.

When the student moves into the registration area the course request and ID cards are presented to the operator. The student number (social security number) is keyed into the system from the ID card. Checks are made on the eligibility of the student to register at this time. A student could be encumbered for one or more of several reasons--having an invalid application date and/or entrance date, or they may have already registered. One might wonder why the latter condition is checked. That is because some who wish only to adjust their schedule will attempt a second registration. Should such adjustments be necessary, the drop and add process administered by the records office must be utilized. The other conditions of ineligibility, as named above, are also administered by the records office at the same service points.

If eligibility problems do not exist, which is true for most, the operator keys-in the student's course selection. The terminal displays the selection and problems related to the selection are highlighted on the screen. A selected course may be full, an invalid call number may have been recorded or keyed, there may be a time conflict, or the same course or section may have been requested more than one time. Again, most students submit a problem-free selection. Nevertheless, the selections as displayed are checked by the student and operator for accuracy of entry.

Following the selection analysis and assignment step, the last screen is displayed. It shows current mailing address, campus post office box number, and questions regarding other cost factors necessary to the proper calculation of student receivables. If housing assignment has been made, this shows at this time. If not shown, the intention of the student is obtained in this regard.

Students may register through the 14th calendar day of each quarter. The process closes for a few days and reopens with students selecting courses for the next quarter. In spring term, students may register for both summer and fall terms upon one trip to the registration station. Our biggest problem here is the preparation of the fall schedule this far in advance. However, this enables us to go to high schools with remote terminals and exercise the registration process before their school year ends.

This concept of continuous registration creates a definite need for decision in regard to payment of fees. Most parents and students are not prepared financially to pay fees in May, June, or July for the fall term. Those who can would just as soon have their money working elsewhere. Problems in this regard are relieved through our billing process as described below.

Except for fall quarter, however, most do in fact pay their fees directly to the business office during the registration period. This helps them avoid lines and secures their selection of classes. Those who have not paid are billed for the fall term to their mailing address and other terms to their campus post office box. Payments may be returned by mail which includes the mailing of their ID card with the payment. When payment is received it is entered by a cashier into the system, the fee card is completed and filed and the validation stamp is placed on the ID card. The ID is picked up by the student at his convenience or returned by mail in a self-addressed stamped envelope.

Billing is initiated some ten days before the first day of class. All who have not paid by the published deadline have their course selection purged. The position of the institution is this: we have provided the opportunity for an early selection of classes relative to needs for graduation. The student must demonstrate the intention of using these resources by making payment or making other satisfactory arrangements at this time; otherwise, the selections are made available to others who participate later. This principle has been well accepted and has served us very well.

Yet, most students go directly to the business office to pay fees. As they do, they locate a fee card that is made available in the lobby area and present it to a cashier along with the ID card. The student ID number is entered and the fee assessment is displayed. These totals are written on the fee card and the ID card is validated for that term. The fee cards are used to produce a cashier control tape of fees collected are are filed as the institution audit trail of the transaction. The control tape of fees collected are balanced the next morning with the daily fee register produced by the central computer. The validation on the student ID card serves as the student receipt and provides evidence of eligibility for student services and/or privileges. Payments by mail, in response to the billing procedure, are processed at the terminals in the same manner.

The question is, how does one get to this stage from the present system? It isn't easy. For example, these concepts were planned in 1967 with implementation targeted for 1969-70. Our first on-line registration occurred July 1975 for fall term of that year. Sufficiently sophisticated computer power and technology must be available and in adequate quantity. The virtual capability of the IBM 370 system provided us with this sophistication on a machine that we could afford. We utilize a 370/135 with 512K bytes of real memory. However, we run with better than 3000K virtual memory system.

Yes, technology must be available and is in many forms and sizes. However, the most important and most necessary ingredients are: (1) an explicit commitment from the top, (2) a designated centralized responsibility with authority equal the responsibility, (3) organized work groups and training programs, and (4) plan for payoff to all involved.

A series of committees and work groups have helped move the ETSU effort to its present level. Included are a policy-level committee consisting of the President, Vice-President of Academic Affairs, Business Manager and Director of Internal Research, Systems and Planning; and two coordinating committees, one for academic planning and the other for administrative planning, this one made up of work groups analogous to the database areas. It is important to have people who will perform the transaction very much involved in the appropriate workgroup.

In summary, while student scheduling is regarded as only a first step, it performs its mission efficiently and cost effectively. The students love the system. It's hard to measure the gain in good will from this fact alone. The system allows us to spread our work activities for a more balanced load. Registration is now just a part of the everyday routine; it does not involve a tremendous, pressure-filled peak for three or four days as in the past. The result is more effective utilization of people, space and facilities.

In short, the system makes everyone involved much more efficient, it provides a much more pleasant environment and does a better job of registering students and providing information related thereto than every before possible.

COMPUTING NEEDS OF A COMMUNITY COLLEGE
ACADEMIC AND ADMINISTRATIVE

Frank A. Kros III
Coordinator of Computer Services and
Information Systems
Delaware Technical & Community College

A. Where we were after 10 years - Brief statement of our status two years ago.
B. Survey of DelTech needs - Results of study of our needs.
C. Alternatives considered - Brief statement of the eight alternatives.
D. Decision - Review of decision process, including approval of Board of Trustees, State Director of Data Processing, preparation of bid specifications and review of proposals.
E. Progress to date - Three of four DelTech Campuses converted to the new system.

A. Where we were after 10 years.

DelTech is a twelve year old, four Campus state-wide system. This paper will review DelTech's computing service.

In 1975 DelTech had an IBM 370-115 computer serving two of its Campuses, a 360-30 computer on a third Campus and a commercial service bureau was contracted to perform limited student record processing for the fourth Campus.

The Campus with the 370 had undergone frequent changes of hardware and suffered from a high turnover of data processing leadership. Student record keeping was virtually the only administrative processing. The balance of the 370 time was used for training data processing students; therefore, there were considerable amounts of idle computer time. The administration of these two Campuses and the President's Office had a high degree of dissatisfaction with the situation.

In contrast, the Campus with the 360 had a record of performance and had the stability of little staff turnover and less frequent equipment conversion. Consequently, more and better information systems were developed. The 360 was also performing centralized payroll and budgetary calculations for the College, as well as, their own student records and other "local" jobs.

The College Data Processing Staff consisted of:

370 Campus:	4 Data Processing Instructors (one of whom was expected to do administrative processing and programming)
360 Campus:	3 Data Processing Instructors plus 2 full-time Data Processing professionals for administrative processing and programming.

There were three completely different systems for keeping student records and no consistent method for reporting enrollment statistics. The College President directed that standard information systems be developed and an Office for Computer Services and Information Systems be created. The President's Office was staffed with 1 full-time individual and 1 part-time individual.

B. Survey of DelTech needs.

The first activity of the newly created Office was to survey the situation and develop a College-wide plan for computer services and information systems.

Priority was given to three Campuses since the 360 system at the Southern Campus was performing adequately. Prior to investigating computer services the College's true requirements had to be determined. It was agreed:

1. Computer services must be available to each Campus.

2. Service should be available to all students and staff (not only the data processing students).

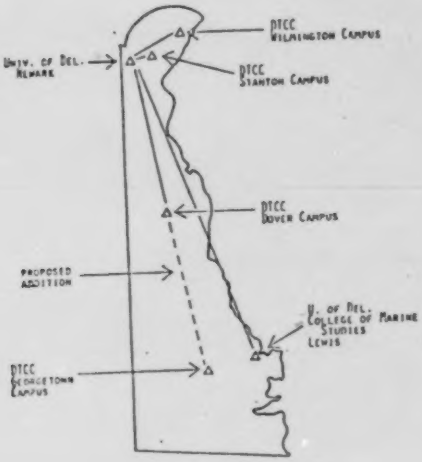
3. Approved user status be granted whenever and wherever possible guided by: demonstrable need, knowledge of operations, and budgetary approval.

4. The users should have direct access to the equipment and be qualified to perform their work themselves.

The College believes it is entering into a new era of computer users. This era suggests the elimination of the "closed door" computer rooms and that all individuals in the very near future should be knowledgeable in order to deal first-hand with the computer, similiar to the knowledge needed to use a telephone, a calculator or to drive a car.

A need for central availability of enrollment and fiscal data was determined. Consequently, a conceptual design of a host system with terminals and satellites at each of the four Campuses was created.

COMPUTING NETWORK



One of the constants having to be dealt with was BUDGET. It was accepted that the budget for equipment and staff would not increase. This meant there would be four full-time people apart from the faculty. The computer budget (not including people) is:

	FY 76	FY 77	FY 78	FY 79 (Projected)
Southern	75,000	90,000	60,000	60,000
Wilmington	92,000	60,000	50,000	50,000
Stanton	20,000	30,000	50,000	30,000
Terry	12,000	30,000	50,000	50,000
President's Office	-0-	-0-	20,000	20,000
TOTAL	199,000	210,000	210,000	210,000

Computing needs were more specifically detailed in three major categories:

1. The College was committed to a data processing instructional program at the Wilmington and Southern Campuses. Providing computer service to this instructional component was given top priority.

2. Other academic areas also needed computer support. There was a very negative attitude about this area of computer service. Computer Assisted Instruction was somewhat oversold in its early beginnings. Despite this fact, the need for academic computing is growing rapidly. Expectations are that this usage will soon exceed all others.

3. Although computers were initally installed for data processing instruction, once available, administrative processing became possible. At present, student record keeping could not be done except by computer.

There appears to be no conflict between academic and administrative usage. With proper planning major conflicts can be avoided. Since academic needs have an understood priority over administrative needs, the administration must work around the academic requirements. The staff understands that serving students also includes keeping good records.

The administration has become increasingly aware of the importance of information systems to the College. The frustration of knowing the computers should be able to assist but in not knowing why they weren't has been an experience for DelTech.

NCHEM provided the blue print for the administrative planning. The College has adopted five data bases as the DelTech master plan. These data bases are:

STUDENT
COURSE
STAFF
FINANCE
FACILITIES

To move from the position at that time to where the College wanted to go seemed a formidable, impossible task. Consequently, a phased plan was developed. Knowing the importance of having recognizable goals at each step of the way, these phases were outlined carefully. Measurement against these goals has given assurance of progress. It is important to make the goals realistic and achievable.

PHASE I

1. *Locate and furnish alternate computer services for the Wilmington and Stanton Campuses.*
2. *Provide communications lines and terminals for the Wilmington and Stanton Campuses to the alternate computer.*
3. *Commence development of systems based upon student and course data, such as registration, grades, class listings, schedules, etc.*
4. *Transfer all usable data currently stored in the Wilmington Campus computer to the alternate computer.*
5. *Terminate the lease on the Wilmington computer.*

PHASE II

1. *Provide terminals and communications lines for the Terry Campus to the computer serving the Wilmington and Stanton Campuses.*
2. *Assist the Terry Campus in implementing systems developed and operating at the Wilmington and Stanton Campuses.*
3. *Such terminals to be available to the President's Office.*

PHASE III

1. *Provide terminals and communications lines for the Southern Campus from the computer serving the Wilmington, Stanton, and Terry Campuses.*
2. *Assist the Southern Campus in implementing systems developed and operating at Wilmington, Stanton, and Terry Campuses.*

PHASE IV

Determine from available alternatives the computer system to be utilized for the forseeable future.

1. *Total utilization of a computer outside the College with terminals to each Campus and the President's Office.*
2. *Total utilization of a College computer with terminals to each Campus and the President's Office.*
3. *Utilization of contract personnel or College employees, or some conbination thereof, for systems development.*

4. Any combination of the above which is determined to best meet the needs of the College.

C. Alternatives considered.

Next, the decision had to be made as to how to get from here to there. DelTech had a number of consultants, including Data Processing Directors from other schools, all of whom agreed that DelTech was expending too many funds for computers and not enough funds for staff. Consequently, idle computer time existed. The College recognized the need to reduce computer expense and increase staff expense.

Having developed a phased plan and after receiving approval of the plan from the President, the Board of Trustees and the State Director of Data Processing, an investigation began for alternatives to implement the phases.

With help from several sources, including the State of Delaware Office of Central Data Processing, a complete set of specifications were developed. These specs were not the normal equipment specs as the problem was not that of hardware. Given the staff and capability to design and program the necessary systems, a turn key solution was required.

Eight different possible approaches were investigated: These approaches were:

1. Use of a 360-30 computer at the Southern Campus.
2. Contract with the University of Delaware for all of the Campuses.
3. Contract with Wilmington City Schools.
4. Contract with UNI-COLL.
5. Contract with New Castle County.
6. Consortium of:
    - -- Delaware County Community College
    - -- Philadelphia Community College
    - -- Widner College
    - -- Delaware Technical and Community College
7. State Central Data Processing.
8. Accept equipment vendor proposals.

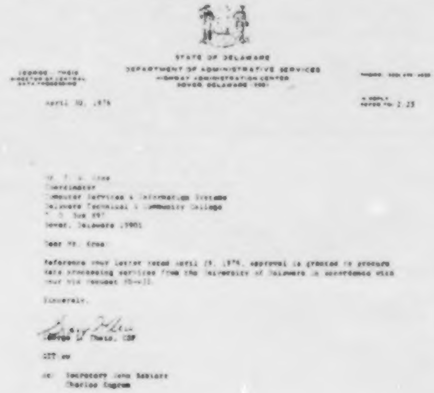
The specs were sent to all and the College awaited the returns. The final specs had, of course, eliminated some of the eight alternatives.

D. Decision.

The decision process should be understood. The overall plan had been presented to and approved by the State Data Processing Director, the Campus Directors, the President and the Board of Trustees. This was a conceptual plan.

Next, the apparent available alternatives were studied. Again, this did not mean final specs, but overall general concepts. Requirements were discussed with many vendors. It was during this discussion that it was concluded that a raw hardware approach was not a viable solution. Obviously, it was ridiculous to think in terms of a host-satellite system with the College's available staff and budget. To keep all levels appraised of exactly what the College was doing, specs were also approved before release.

When the University of Delaware Computing Center was chosen as the only real answer to the College's requirements, approval was readily obtained. The State approval took one day.



Factors considered in arriving at the recommendations:

- The need for the College to have a large staff of computer specialists, primarily programmers to maintain and enhance the software supplied by the manufacturer, should the College maintain its own computing center.

- Operating costs, just as capital costs, must be calculated for all expenses arising from the computers presence, not just for those directly attributable to its operation. Labor costs, for instance, must allow not only for the salaries and benefits of staff employed directly by the computing facility, but also a proportion of those employees indirectly involved; i.e., general maintenance staff, security personnel, and so on.

Since technological developments have reduced hardware costs at the same time that personnel costs have risen, it is important not to overlook indirect costs.

Major advantages of joining with the University of Delaware Computing Center are as follows:

- -- Provides ready access to a variety of special computers and increases computer versatility.
- -- Allows DelTech to satisfy more of the computing needs which exist on each Campus (instructional, administrative, community services, etc.)
- -- Allows the College to specialize in instructional computing (computer assisted instruction, data processing instruction, computer managed instruction, social science research, etc.) without creating a specialized and a generalized computing center.
- -- Provides fixed cost for each contract, regardless of usage; therefore, ease of budgeting.

Further advantages of the contract is that the University of Delaware Computing Center agreed as part of the contract to:

- -- Develop a Student Record System to meet needs.
- -- Install Remote Job Entry equipment and terminals. The contract provided for lease-purchase of this equipment.
- -- Develop a state-wide communications network for DelTech.
- -- Continue application and technical support to DelTech throughout the life of the contract. This enables the College to operate with non-data processing people at each Campus.

E. Progress to date.

In August 1976 RJE equipment and terminals were installed at the Wilmington and Stanton Campuses. The student records were transferred to the University of Delaware Computing Center and the new programs were operational. The 370 was removed.

The Student Record System was designed by a University of Delaware task force of computer systems and programming people. These people were dedicated professionals who worked long and hard to accomplish this task. DelTech retained its disk packs of data and programs, ready to go to a backup site if necessary. The Terry Campus requested to be brought into the system ahead of schedule. This was an endorsement of our success.

Soon after installing equipment at the Terry Campus, additional terminals were ordered from Campus funds and the College approved the establishment of a Data Processing Instructional Program.

At each Campus, staff training projects were established for faculty, student services and business office personnel. Presently, 250 DelTech staff members have been trained and are using the computer.

There are many aggressive users among the non-data processing faculty at each Campus. As instructional staff gain more experience and discover the vast amount of available material, this usage continues to increase.

At this conference, Drs. Boas, Frantz and Matthews of the University of Delaware are reporting on a major staff training project. The results of this training is electric. Instructors who never went near the computer two years ago are now avid users.

Administrators are also sharing in the usage. No longer do the Registrars wait for a data processing professional to run the computer for them. Each Registrar and staff do their own work. Complete instruction was provided along with on-site support during the first registration period which enabled these people to be self-sufficient, except for occasional assistance when trouble occurs.

Business Office personnel likewise have learned to use the system. They have programs to perform tuition accounting, financial aid accounting, part-time employee time-keeping and gross pay calculations.

DelTech has gone from a College with nine or ten people using the computer to more than 250 staff with user numbers able to use the computer. DelTech's major problem at this time is the lack of terminals. The College has gone from no terminals, 18 months ago to:

Wilmington	4 Terminals
Stanton	5 Terminals
Terry	12 Terminals
President's Office	3 Terminals
Southern	3 Terminals

At this time the Southern Campus is looking carefully at the situation on the other three Campuses. No decision has yet been made regarding their 360 computer. There is a College-wide study underway. The three Campuses have elected to continue with the University of Delaware. However, DelTech will continue to review this periodically. DelTech is aware of the rapid changes now taking place in computer architecture. Distributed processing, micro processors, decreasing hardware costs, all will have an impact on future plans.

My forecast as to the results of this study would be that DelTech will probably opt for increased local processing while retaining a link with the University of Delaware Computing Center.

EXPERIENCES
WITH A
COMPUTER ADVISORY COUNCIL

Sister Patricia Marshall
Xavier University of Louisiana
New Orleans, Louisiana 70125

When Xavier University of Louisiana acquired a computer in 1968, there were high hopes that computing would become a change agent in both administrative and academic systems. In time these expectations came true, and intense struggles swirled around computing and around the money to fund its development. Playing a role in these happenings was a computer advisory council, forged in 1975 following six years of effort. This paper addresses the way the council was organized, how it functioned, and the results of its decisions, as well as the roles played by other forces. The paper is of general interest to academic and administrative personnel in higher education, as well as to anyone concerned about the impact of computing on an organization. Though long-range results are still in the future, Xavier's experiences may serve as a starting point for others of what to do/what not to do in a particular situation.

When Xavier University of Louisiana made its quantum leap into the computer age in 1968, we brashly voiced the hope that computing would become a change agent in administrative and academic systems. Little did we sense how profoundly that hope would be realized and how intensely the struggles for control and power were to center around even the first hardware configuration. We know now that we set into motion a dynamic complex of continuing change, not only in the way we do things, but also in the way we perceive management of the computer services and the process of change, and in the way we perceive budgetary allocations and priorities.

BACKGROUND

As members of a regional group of small colleges and universities experimenting with various forms of computer equipment for instructional purposes under the supervision of Dr. John Hamblen (then with Southern Regional Education Board), we began with a small stand-alone computer financed in part by the National Science Foundation. Xavier, whose student body then numbered only about 1300 (600 fewer than today), was too small to justify a computer for academic purposes alone (this was prior to the eras of mini- and micro-computers); so from the outset administrators also used it (initially for registration, student billing, and student records). We made vigorous efforts to educate and attract student and faculty users. These efforts included the offering of courses in programming under a "developing" computer science department. At the same time, efforts were made to ease the transition for the registrar and fiscal director by assisting with data preparation and pointing out what else could be done if personnel existed.

Though we had carefully placed the computer center organizationally where the director would report to the executive vice-president (thwarting an attempt on the part of the fiscal director to gain control from the start), it became evident early on that academic users in particular needed strong representation and more help. The functions of heading a "developing" computer science department, as well as of giving leadership to academic computing, were combined with directing the computer center. All of these hats were worn by one person, who gradually came to be perceived by both academic and administrative personnel as extremely powerful, although the responsibilities were rarely backed up by authority.

Upgrading of the initial hardware, occurring several times during the period from 1973 through 1977 (and accomplished mostly with academic funding), assisted us to grow in the ability to serve users; but at the same time it also gave impetus to a "closed-shop" situation in which users could no longer descend bodily upon the computer, push their own buttons, and "blow" their own discs. The frustrations of dealing with a non-hands-on batch system drove away some academic users until we also acquired a small time-sharing system in 1975 with terminals for the academic users (by piecing together donated equipment and academic grant funds from here and there).

On the administrative front a fiscal programmer, hired in 1970 to "get things on," did just that, with so much pressure to "hurry up" that journals, ledger, payroll, etc., were designed with no thought for producing summary-type reports for Management (the programs were all written for accountants). Meanwhile, Management, needled by the Board of Trustees (and a consulting firm it had hired), realized it was not getting the types of reports from the fiscal department that it needed. Several changes of management in the fiscal department occurred, and computer center personnel lost sleep, hair, and a scalp or two in the effort to keep pace with changes in philosophy and the chart of accounts.

The consulting firm hired to assist the fiscal department recommended a larger computer. The Board of Trustees agreed and told us how easy it was to get one, but they were unable to find the additional financial resources needed.

Meanwhile Xavier received a grant which included funds for programmers, analysts, etc. Again the Board of Trustees got into the action, effectively blocking the hiring of some of these. The reasoning was that the consulting firm had recommended purchasing software (their own, which was not compatible with our computer), and therefore we did not need analysts, etc. Meanwhile, all student record-oriented systems development (admissions, new student information system, alumni, etc.) stepped aside while Fiscal got top priority.

COMPUTER ADVISORY COUNCIL DEVELOPMENT

Not long after we first acquired a computer, Dr. John Hamblen, acting as consultant to the principal investigators on various campuses of the small-college experiment, offered some thoughts on a computer advisory committee by way of a memo in which he stated: "First let's agree on 'advisory to whom.' Certainly not to the person who is responsible for the day-to-day activities of running the computer center; i.e., not the director. If the college administration feels the need for an advisory group on computers, then this group's activities should be as follows:

> "to advise and make recommendations to the administration on the level of support to be given to the computer center
>
> "to advise and make recommendations on broad operation policies of the center (priority systems, hours of operation)
>
> "to serve as a sounding board for determining the general condition of the relationship between the center and its customers
>
> "to advise and make recommendations on the selection of major equipment

"If a panel is to be able to perform well in the above areas, it is obvious that they must be well informed with regard to the activities of the center. This group should be appointed by the president upon the recommendation of the computer center director. The director should prepare his list of recommendations from the largest users of the major areas served, such as instruction, research and administration.

"If selected carefully and utilized properly, an advisory group can be a useful ally to the computer center director by supporting his case for adequate staff and facilities to provide the services his customers need.

"A suitable chairman of the advisory group might be the person to whom the director reports. The director should be a member of the committee or at least be present at all meetings of the group.

"These are general statements as to my feelings on faculty advisory groups for college computer centers...In a given case, I might modify or amplify some of my statements..."

We sent the Hamblen memo with one of our own to the executive vice-president in February, 1969, suggesting that he might chair the group himself at first. The recommendation was considered, but implementation was delayed indefinitely. Meanwhile the years slipped by, and budgets, five-year plans and an ever-growing variety of problems received only cursory attention in the press of day-to-day operations. However, we did take the initiative to separate out a computer science budget, seek a separate cost center number for computer science, and keep separate files in the hope that the computer center director would eventually become "unbundled."

Highly critical memos began to emanate from the chemistry department (which had been our largest academic user), and by the fall of 1974, a petition from that department was circulated requesting the formation of a "users' committee," but with significant differences from the functions recommended by Dr. Hamblen: it called for decision-making power on policies; it specifically excluded Computer Science (a user) and included Biology (a non-user); it assumed the chemistry department would form the group, set up the ground rules, etc.

We resurrected Dr. Hamblen's memo, requesting again that the executive vice-president consider setting up an advisory group, though along the lines originally proposed; this time we sent a copy to the dean. To ensure our own credibility, we mentioned candidly some of the ramifications: users would probably complain about the inadequate air-conditioning system, request programming assistance, ask for a second shift, etc. At the same time, we suggested that the vice president might want to chair the group.

Approximately one year and several dozen memos later, the computer center director became "unbundled" (someone was hired for computer science and academic computing), and the dean joined forces with us toward moving in the direction of an advisory group. We pointed out that we were now well into a second shift, and that the Board of Trustees was proposing a much larger system than we could afford. We tried to show that the need for planning was not satisfied by the existing planning structure; a computer advisory group could focus the fragments of information into a comprehensible picture; and such a group could even provide information for other decisions involving related areas such as maintenance, space needs, proposal writing, etc. We also cited the following additional advantages: the formation of an advisory group would enhance our image with funding agencies (where grants involved computers); it would show the Board of Trustees we were not taking their suggestions lightly; it would show disgruntled faculty members the administrators do care about their problems; it would foster communication, especially among administrative users.

The group was formed by November, 1975. It was

named a permanent council (rather than a committee), and it began to hold its two meetings per semester, with subcommittees for academic and administrative users meeting more often. A copy of the constitution is appended.

The two subcommittees have been extremely active. The administrative subcommittee was chaired first by the planning director and later by the administrative services director. Most of the time it has met weekly. The executive vice-president and the computer center director have been ex officio members. The computer center director has acted as secretary, circulating minutes among all council members (including academic). This open communication policy cost the computer center a potential employee: the social sciences almost bound their research director in chains when they discovered one of the administrative people had recommended he be hired by the computer center as programming director. On the other hand, the academic users feel it has kept them informed better than anything else on what the administrative users were planning.

Branching off from the administrative subcommittee in the fall of 1977 was a group involving more potential administrative users: the SISDT, or student information system development team. Another offshoot is a special subcommittee to deal with privacy of records and security, whose charge it is to develop policy for top administrators to approve. Each group circulates its minutes widely.

The academic subcommittee, chaired by the academic computing director, also circulates its minutes among all council members. The executive vice-president and the computer center director are ex officio members but do not often attend. Unfortunately, but predictably, this has served to reinforce the notion that "the administration" doesn't care about academic users and that the computer center director's sympathies lie with the administrative users, even though the operations manager regularly attends the meetings.

A couple of academic users belong to both committees to cross fertilize the discussions.

In the first year of the council's existence, the academic users gave brilliant summations of their positions, while the administrative users appeared in almost total disarray. Using flip charts, the academic subcommittee presented projections of future usage and justified the request that the small time-sharing system be dedicated entirely to academic use. The motion carried. In praising the academic subcommittee for its work, the executive vice-president inadvertently used the word "slick" to describe their presentation. Academic users felt there was something other than praise between the lines, and the schism between academic and administrative users grew wider.

In the second year, 1976-1977, the Council rarely convened, but battles raged at the subcommittee level during the entire year. During one meeting of the large group, the executive committee proposed that a task force be formed to prepare a long-range computing plan for the university. Because members were unable to agree, either on the length of time or the makeup of the proposed committee, the plan was voted down. Frustrated by the apparent rebuff, the executive vice-president decided to proceed with his plans unencumbered by the larger group. Accordingly he hired additional consultants (who, being from the industrial sector, were fiscal oriented) and authorized experiments by one of the administrative users, seeking help from another consultant (this time a university-oriented person) to carry it out. The experiment involved the use of a video terminal and the establishment of a remote-job-entry link between our computer and TUCC (Triangle Universities Computing Center) in North Carolina. The assisting consultant, in an effort to show the administrative users involved that they could manage all this by themselves, did not consult any of the knowledgable technical people or other administrative users at Xavier. Problems with the installation of equipment surfaced, however, and additional concerns about privacy of information arose.

When questioned at an administrative subcommittee meeting about what was going on with TUCC, the administrative users involved did not wish to comment. This deepened the growing schism between two groups of administrative users, the fiscal oriented and the student records oriented.

As the apparent game playing intensified, administrative users other than fiscal began to speak up. Concern was voiced that the fiscal department might, in its search for faster information retrieval, make even more difficult the establishment of software bridges or interfaces among systems to facilitate planning and institutional research.

Meanwhile a consultant was preparing an "evaluation" of Xavier's computing designed to show that Xavier's computing needs could best be met by a link with TUCC. This would be in accordance with the "utility concept," as he called it: a utopian world in which administrative users would need little in the way of programming services but could sit down at terminals and get information.

Academic users, sensing a gigantic decision was in the making, and afraid they might be forgotten in the shuffle, prepared a proposal of their own (a university-wide proposal was about to be submitted to a funding agency). Meeting several times, they hammered out their requirements, looked exhaustively at several alternatives, and finally arrived at unanimous support for an in-house time-sharing system that would meet their well-documented needs (authoring language for CAI, BASIC, FORTRAN, etc.). They backed up their proposal with the advice of a consultant, who pointed out in his report that the model proposed could be upgraded and used for administrative work as well.

Administrative users, aware of this development from the minutes, and motivated by the impending deadline for the proposal, held an emergency meeting. At this meeting the computer center director presented usage records showing comparative usage by large users, trends, etc.; a large array of alternatives (including costs, advantages and disadvantages of each), a summary of the problems with administrative system development and what we had learned from them, and a documented summary of projected on-going costs if we were to put all administrative work at TUCC. Later the personnel director made estimates of his own which were much lower, and the computer center director countered with some case histories from within the TUCC network which supported the original estimates.

Meanwhile the fiscal director became aware that the in-house time-sharing system proposed by the academic users was the same system he had once investigated and found would meet his needs. Recognizing also the importance of working together, he changed his position. He and all the other administrative users then voted for an upgraded model of the system proposed by academic users.

Then came the day of the full council meeting. Minutes before, copies of the report recommending the link with TUCC were circulated among some of the members. Reports and motions were presented, and we reconvened in 24 hours to vote. Except for two persons, who had not attended meetings or read the minutes, all voted to acquire the system proposed by the academic users' group. The executive vice-president expressed gratitude for the unanimity, and for a few days all breathed deeply from the atmosphere of mutual trust and good will.

As it turned out, purchase of equipment was not encouraged by the funding agency; other priorities crowded into the proposal; and politics dictated a different course of action. However, the impact of dozens of meetings would be felt for a long time. Administrative users know they must work together. Academic Computing has established an identity of its own and cannot be ignored. The executive vice-president is more aware of how long it takes to develop a system and of the need for executive attention to computing.

Of course all is not quiet on the Xavier front. The past may pale in comparison with the future. No doubt more lessons will have to be learned. But the computer advisory council will continue to be "where the action is."

CONCLUSIONS

Taking Hamblen's recommendations as norms, one can spot most weaknesses and strengths of Xavier's experiences with an advisory group. The administration was somewhat better informed about computing needs after two years. Academic voices were heard to a much greater extent than before, and administrative users had begun to cooperate with each other more than they had previously done. The need of many users for remote video display came through clearly.

In addition, more information became available about university-wide computing requirements for planning purposes, and responsibilities once bundled in one person (computer center director) were now shared, resulting in better service.

On the negative side, not all communication problems were solved (despite tremendous progress). Possibly an earlier formation of the group would have helped.

On balance it can be said that Xavier is far better off with the advisory council than without it, and the increasing attention paid to computing by the executive vice-president as a result of the experience should pay off in the long run.

APPENDIX

CONSTITUTION
COMPUTER ADVISORY COUNCIL
XAVIER UNIVERSITY OF LOUISIANA

ARTICLE I. NAME

The name of the body constituted in this document shall be the Computer Advisory Council of Xavier University of Louisiana. This organization is hereinafter referred to in this document as "the Council."

ARTICLE II. PURPOSE

The purpose of the Council shall be:

(1) to advise the Office of the President of the University and make recommendations on the selection of major equipment, space, etc.

(2) to advise and make recommendations on broad operation policies of the Computer Center (priority systems, hours of operation, chargeback systems, etc.)

(3) to advise and make recommendations on the level of support to be given to the Computer Center

(4) to serve as a sounding board for determining the general condition of the relationship between the Computer Center and its customers.

ARTICLE III. MEMBERSHIP

Section 1. Initial members

Initial members shall be appointed by the President upon recommendation by the Executive Vice-President, the University Dean, and the Computer Center Director, on the basis of their being well informed on some aspect of computing, as well as being capable of representing others. These members are listed in Appendix A.

Section 2. Additional members

If additional members are desired by a simple majority of the existing membership, they shall be added by presidential appointment.

Section 3. Filling vacancies

Vacancies shall be filled by presidential appointment.

Section 4. Length of term

A. Membership on the Council shall be permanent except as otherwise indicated herein.

1. Faculty members shall serve a term of three years and may be reappointed.
2. Students shall serve a term of one year and may be reappointed.
3. A member may be recalled and replaced by the President if a simple majority of the membership makes such a recommendation.

ARTICLE IV. OFFICERS

Section 1. Officers

A. The officers of the Council shall be a Chairperson, a Vice-Chairperson, a Secretary, and a Parliamentarian.

1. The Chairperson shall be the Executive Vice-President.
2. The Vice-Chairperson shall be the University Dean.
3. The Secretary and Parliamentarian shall be elected by a simple majority of a quorum at the first regular meeting for a term of one year.

Section 2. Duties

A. The Chairperson shall preside over all sessions of the Council and the Executive Committee. He/she is an *ex officio* member of all committees of the Council and shall represent the Council on all occasions where representation is required.

B. The Vice-Chairperson shall assume the office of the Chairperson when the Chairperson is absent. He/she is an *ex officio* member of all committees of the Council and shall coordinate the activities of all committees of the Council.

C. The Secretary is the official recorder for all sessions of the Council. He/she shall keep proper records. She/he shall distribute copies of the minutes of Council meetings to all members. He/she shall maintain a current roster and shall keep the record of attendance. He/she shall distribute notice and agenda of a Council meeting ten days prior to the date of the meeting.

D. The Parliamentarian shall have the duty to interpret this constitution and to see that the Council meetings are carried on in an orderly manner and in keeping with proper parliamentary procedures.

ARTICLE V. MEETINGS

Section 1. Regular meetings

There shall be two regular meetings of the Council during each academic semester. Normally, these shall be held in mid-September, November, February, and April. Actual dates, places and times shall be determined by the Executive Committee.

Section 2. Special meetings

Special meetings of the Council may be called

by the Chairperson of the Council with the consent of the Executive Committee

by the Chairperson of the Council at the request of at least five members (and with the consent of the Executive Committee).

Section 3. Quorum

At least ten members shall constitute a quorum at a Council meeting. All members shall have one vote apiece.

Section 4. Rules of order

Unless otherwise provided in this constitution, the rules contained in Robert's Rules of Order (latest edition) shall govern the procedures at, and the conduct of, the meetings of the Council and its committees.

ARTICLE VI. COUNCIL COMMITTEES

Section 1. Executive Committee

The Executive Committee shall consist of the officers of the Council. It shall be the duty and responsibility of the Executive Committee to carry on and supervise the affairs and the business of the Council between its regular meetings and to deal with the business of the Council in general, including preparation of the agenda and selection of dates, times and places for meetings of the Council. The Computer Center Director shall be an ex officio member of the Executive Committee.

Section 2. Special Committees

Special committees may be appointed by the Chairperson from time to time to deal with any special or specific matter. These committees shall have the power and duty to make studies and to investigate the matters assigned to them by the Council, and to make recommendations and give opinions to the Council in that regard.

Section 3. Standing Committees

Standing committees shall be formed as the need arises to deal with on-going sub-areas of concern such as

academic computing
administrative computing*

Section 4. Appointments

Members of a special or standing committee will be appointed by the Chairperson of the Council. At the recommendation of the majority of a committee, non-council members may be appointed to a committee by the Chairperson.

Section 5. Reports

A special or standing committee shall make written reports to the Council on any matters it has been charged with studying. Standing committees shall circulate copies of minutes to all members of the Council.

ARTICLE VII. AMENDMENTS

Section 1. Proposals

Amendments to this constitution may be proposed by a resolution adopted by a majority of those present and voting at a meeting of the Council, provided that number constitutes a quorum.

Section 2. Adoption

Amendments shall become effective upon ratification by two-thirds of the membership of the Council.

ARTICLE VIII. ADOPTION OF THIS CONSTITUTION

This constitution shall become effective upon adoption of it by two-thirds of the membership of the Council.

---

* (Other sub-areas were named originally which might have centered around hardware, but these were never activated.)

DETERMINATION OF DATA PROCESSING REQUIREMENTS AND EQUIPMENT IN A MODERATE SIZE STATE UNIVERSITY

John J. Murphy
Indiana State University

The subject paper presents, for a state supported educational institution, a framework for organizing a study to identify the various electronic data equipment user and non-user data and information needs, assigning priority to those needs, and the overall institutional education effort requirements. Outlined also are necessary considerations to be made in evaluating alternative computing systems, their utilization, response to identified needs, and future growth. Incorporated is a discussion of long-range strategic planning for future developments and uses. It is anticipated that the result will be a guide to the typical state-supported university contemplating either the upgrading or initial installation of a data processing system.

INTRODUCTION

A major difficulty in the selection of data processing equipment in both private industry and educational institutions is the attainment of concise user requirements. Even after the establishment of such needs, how they are to be met in terms of equipment and within budget limits is controversial. The root cause of much of the discord in a university environment is the conflicting objectives of: first, the administrative organization as opposed to those of the academic structure, and; secondly, the differences within the academic community itself. Administrative requirements tend to be traditional ones--student and faculty records, accounting documents and reporting, information compilation and summarization to various governing bodies and agencies, statistical data for planning, budgeting, and control, and routine clerical duties.

In contrast to administrative specifications, academic needs are quite varied and criss-cross a wide variety of disciplines and user backgrounds. The primary objectives of the academic units are instruction utilization and research applications. However, even this breakdown is inadequate. Instructional use can vary from that of establishing a detailed technical and working knowledge of the equipment and concepts to that of an informed user concerned only with occasional use and its potential applications and limitations. Research needs are no less varied. These needs can require on-site experiment monitoring and control to supporting data analysis and reporting. The task of sorting through these needs and establishing their priority is paramount to the establishment of a successful equipment configuration and equally as important, the data processing group's organizational structure.

Another problem that must be dealt with is that of the "non-user" community within the institution. The proliferation of data processing in the past two decades has left many of us using it with the false impression that everyone is equally receptive and knowledgeable. Quite the opposite is the case, however. The importance of educating ourselves to and assessing the "non-users" needs is critical on several counts. First, such a group usually is most vocal in their arguments for alternative budget/priority choices. Secondly, if this group is quite large and aggressive, they can impose a less than adequate system on the active users either via under-allocation of funds or ultimately becoming users causing unplanned needs growth that tax the final systems design potentially rendering it useless.

THE PRELIMINARY STUDY PHASE

The primary objective of the preliminary study is to establish a perspective of the overall institution's needs and requirements. This aim will be accomplished by surveying the various academic and non-academic areas to identify existing and potential applications for an electronic data processing system. The initial study is not meant to be a substitute for a subsequent full scale and detailed analysis but rather a vehicle for defining the overall goals for such a study.

It is initializing and conducting such a preliminary study that the novice or inexperienced institution often encounters its greatest difficulties. A major pitfall is that once begun the initial study will be expanded beyond its original objectives and attempt to accomplish too much. Under such conditions the entire concept of data processing evaluation and upgrading could be jeopardized as parochial interests are advanced to the detriment of institutional needs.

All too often an institution will compromise too quickly its data processing potentialities by limiting or focusing its initial analysis on a few established or vocally expressed applications and by ultimately installing a system based only on such limited objectives. If an institution does not carefully study all of its organization, it has no way of telling whether the equipment it acquires on limited information is adequate for future application and expansion or not. The only way to avoid this problem is to initially examine all potential applications for their present and future feasibility. If it appears likely that, at a later date, it will be worthwhile to add other tasks to the system load, this obviously should be a factor in the specification of the systems design.

It is not unreasonable to plan for a piecemeal approach to equipment acquisition and a gradually-expanding facility. The danger is that such a program would be haphazard. The information provided by the initial study can be used to establish the conditions and the parameters for an orderly and efficient growth schedule, and to insure that the system design expands in a timely fashion as appropriate to the institution.

The preliminary phase study is also an excellent way to raise the "awareness" of the institution's various academic personnel and administrative management. In conducting the initial study, the emphasis should not only be on gathering facts but on their dispersion also. In this fashion, everyone will gain confidence in the objectives of the data processing system and will at the same time gain a familiarity with the interdisciplinary dynamics of the institution's operations.

THE PRELIMINARY STUDY GROUP

There is a tendency during the initial study of needs of a multi-discipline institution to bring together all those units or individuals that might be remotely affected. Such an attempt is unnecessary at this stage and quite often will only retard the evaluation process. A large team of people is just not required nor even desirable. In fact, the team needs to be kept as small as possible; a manageable number would be in the range of three to five individuals.

The individuals comprising this group should be broadly disciplined. A person need not be particularly expert in data processing to qualify, but should more importantly be open-minded and able to comprehend the multi-faceted composition of an educational organization. While mandatory that at least one member be thoroughly versed in recent technological developments in the field, the remainder will be the more valuable for their ability to recognize or to identify potential applications and interactions. This ability must transcend applications already in place to those that potentially exist.

Other advantages of a small group over a large representative committee are the opportunities to perform tasks expediously and to provide organizational flexibility. A small group is also conducive for the assignment of responsibility and accountability. It cannot be emphasized too importantly that to allow this group to grow due to political expedience, individual academic unit's aspirations, or just on the false sense that everyone must be represented is self-defeating. This is a task group. In the process of its study, the team should provide alternative mechanisms for each unit or individual to be heard and to contribute to the overall systems specification.

ADMINISTRATION/FACULTY PARTICIPATION AND EDUCATION

Those individuals among us that have technical and data processing backgrounds tend to ignore the importance and impact of behavioral considerations on the introduction and use of data processing equipment. We tend to think, respond, and perceive data processing utilization as so rational an approach to the solution of so many of our problems that it is inconceivable that everyone else cannot. In our minds such behavior is irrational and illogical. Whatever its cause, however, it presents formidable roadblocks if not dealt with effectively.

Many data processing applications and expansions fail because of inflated user expectations and failure of the designers to account for the political environment. Users become disenchanted with EDP installations because their applicability was raised beyond reasonable limits in attempts to "sell" the concept and because prospective users were not fully articulating their expectations. These problems are particularly acute among first-time users and individuals unaccustomed to concrete formulation of objectives and measurable results for data processing.

Another hinderance to data processing applications is the perceived restrictions it places on the political structure of the organization. The "rational" measure and accumulation of data and information often restricts, at least in the administrator's or faculty's mind, their flexibility. To such an individual, all arguments fall on deaf ears. He does not see the information as increasing his knowledge or "reach" but as an obstacle to his normal political behavior.

The primary tool with which to deal with these obstacles is education. Surprisingly this is an area that quite often is neglected in an academic environment. Too often it is assumed that because the installation is in a learning institution that most if not all those potentially affected will automatically assume responsibility for becoming familiar with the idea. Nothing is further from the truth. The pressures of time and conflicting demands and objectives weight as heavily on the members of academic institutions as on members of industry. What is necessary, therefore, is a number, between three and four sessions, of short informative seminars to orient the faculty and staff. Such seminars should be offered to each of the major academic and administrative units and cover the potentials and drawbacks of EDP. Materials to be discussed might include computer concepts, systems-analysis techniques, project evaluation, computer-hardware installations, and computer-system management.

One more precaution. Such educational seminars will be ineffective if not carefully planned and executed. More often than not those already familiar with the concepts will attend. There must be some high level commitment made by the top administration and this concern conveyed throughout the institution to get the "Cassandras" and "Nay-Sayers" to attend. It may be necessary to involve such top-ranked individuals in the actual seminars or in some other highly visible manner. Participation in this fashion by the administration is beneficial because it adds credibility and signals the organization as to what the future trends will be in addition to showing commitment to the concept.

Once initiated, administrative participation must be continuously encouraged, sought after, and incorporated in the study. This is part of their educational process and will aid in the clarification and presentation of the feasibility study results. Such participation reduces the "black-box" mystic surrounding data processing and attracts others who may not become involved if it were not for the example set by their superiors and colleagues.

CONDUCTING THE PRELIMINARY STUDY

Once the organizational and educational backdrop for the preliminary study has been completed, the actual study can be set in motion. The study should begin with the setting of its objectives and scope. The time spent and detail incorporated into specification of the study objectives will determine the nature of the activities throughout the remainder of the study. Failure to establish objectives or to do so poorly can, and most likely will, result in misstatement of needs and resource allocations that shall plague the study continuously and its resulting recommendations.

Some thought at the initiation of the study should also be given to the manner in which the information gathered will be recorded. A consistent and easily understood format should be established from the first. This approach will be helpful in at least two ways: first, it will provide a guide for gathering pertinent data, and; secondly, the resulting information will be relatively homogenous and traceable for follow-up analysis. Such documentation is vital and not a luxury to minimizing the poor utilization of the study group's resources, the retracing of steps, and the quieting of any controversy over interpretation of the results.

Typically the data gathered and documented for both user and non-user groups would be their background, goals and objectives, the major policies and procedures with which they function, any regulatory rules or restrictions they must comply with, and ultimately any inputs, outputs, and/or resources used in the performance of their objectives. The emphasis at this stage is the formulation of user specifications for the data processing system and every attempt should be made to downplay or avoid specifics in terms of hardware. The study group will find that existing users will attempt to formulate their needs in the form of specific hardware and while this is initially tempting to accept it defeats the purpose of an overall responsive system design.

Conduct of the preliminary study can assume many various forms among which are review of each groups internal and external records, interviews with faculty, staff, and administrators, and work-sampling to establish the flow and volume of existing or potential use within a particular group. The appropriateness of each technique depends on factors a few that might be summarized as follows: 1) the purpose of the application such as an exhaustive systems overhaul or a more limited reaction mechanism for crisis, 2) the time horizon for study which may either limit the data being acquired or impact the composition and commitment of the study group itself, 3) institution and individual unit attitudes and cooperativeness, and obviously, 4) the quality and quantity of existing materials and documentation. Once such factors have been assessed, the study group is in a better position to determine the trade-offs between resource expenditures and information to be gained by the application of each technique.

During the preliminary study, attention should also be directed at understanding and documenting the political and behavioral interactions within each group as well as among groups. Understanding of these factors will explain many of the deviations from established procedures while shedding light on arguments for or against a proposed system or specific hardware configuration. In an academic environment, the political considerations will assume greater importance as the proposed data processing system moves from the planning stage to the implementation stage.

EVALUATING PRELIMINARY STUDY INFORMATION

Once the study group has completed the data gathering process, a list of the existing and/or potential computer applications can be developed. A suggested approach to this phase is the use of a 'specification matrix' similar to the simple example shown in Figure 1. The utilization of a matrix will quickly identify similar and interrelated applications and help eliminate duplications. The actual matrix used would be much more detailed than that shown and include information with regard to data base design, frequency of application, form of system entry and retrieval, formats, etc.

FIGURE 1

SPECIFICATION MATRIX

Application Functional Area	Hardware Instruction	Hardware Design	Programming Instruction	Business Applications Instruction	Mathematical Applications	Accounting Records	Student Records
Academic Affairs						X	X
Student Affairs						X	X
Physics	X	X	X		X		
Engineering	X	X	X		X		
Dean of Arts and Sciences						X	X
Business			X	X	X	X	
Mathematics		X	X		X		

Identification of tasks and applications paves the path to selection of alternative hardware configurations. By taking advantage of the matrix analysis approach differing specification groupings and solution methods can be hypothetically configured and priced out. Here a "hardware matrix" as illustrated by Figure 2 might be applied. Displaying the service requirement and hardware solutions will aid in establishing a range of potential configurations from a "bare-bones" to

FIGURE 2

HARDWARE MATRIX

Application Hardware/Software Utilized	Hardware Instruction	Hardware Design	Programming Instruction	Business Applications Instruction	Mathematical Instructions
CPU	≥ 64 K	≥ 64 K	≥ 128 K	≥ 64 K	≥ 64 K
Cardreader	X	X	X	X	X
Printer	X	X	X	≥ 1000 LPM	≥ 1000 LPM
Magnetic Tapes	X	X			X
Paper Tape	X	X			
CRT					X
Magnetic Disks	X	X			X
RJE					
Remote Teletype	≥ 60 CPM	≥ 60 CPM	≥ 80 CPM	≥ 80 CPM	≥ 60 CPM
PL1			X	X	
COBOL			X	X	
FORTRAN			X		X
BASIC			X	X	X

staisficing to optimum and their corresponding prices. Because most educational institutions are expense constrainted and non-profit, the emphasis will be on cost minimization for given levels of servicing. The cost listing and specification matrix, therefore, will be the two most formidable tools the study group may develop in support of the implementation of changes in the existing data processing system or of the introduction of one to the campus.

From the specification matrix and cost listing, the administration will derive a fairly comprehensive picture of the basic EDP system needs of each academic and administrative functional area, their interrelationships, and possible solutions. At the same time, the preliminary study should be presented to the various affected groups for their comments and as a means of making them aware of the overlapping and the individualized needs throughout the institution. Administration now has some feel for total dollar investment required and is in a better position to assign a priority to conflicting data processing needs consistent with its overall capital investment program and startegic planning. At this point the only concrete decisions that can be made are whether or not to proceed or reject the concept of improved data processing and the target investment commitment should the decision be to proceed.

THE DETAILED STUDY

If the decision is to continue and implement a data processing system, the preliminary study is a starting point and should not be considered adequate for final configuration specification. What is required now is an intensive in-depth effort to arrive at an optimum systems design within the budgeting constraints suggested by the administration as a result of their study of the preliminary report.

Each component of the detailed study has a correspondent part in the preliminary study. The major difference between the two studies will be thoroughness with which the detailed study is conducted. One other major difference will be with expansion of the study group. This group should now be expanded from the original three to five individuals to at least one person from each user and non-user group and/or functional area. The original members can serve as a cadre and coordinating body while the new members utilizing the preliminary study as a guide can be given the responsibility to go over their particular area clarifying, deleting, elaborating, and verifying the original material. These efforts should validate for each functional area the accuracy of the original specifications and refine these to concise statements of needs.

As done in the preliminary study, the resulting functional area specifications should be placed in an expanded specification matrix and hardware matrix to identify overlapping requirements and to eliminate duplications of hardware. Once summarized in this format, various hardware manufacturers, computer service operators and equipment lessors can be consulted as to how they might comply with the resulting specifications given the target budget. If at all possible, an independent computer consultant should be requested to review the study material and offer suggestions at the same time. The utilization of such an expert may be more than worth the price of the service in the savings that result in preventing the institution from being overcome by endless array of equipment and services offered by these firms. Also consult with other institutions who have or are going through similar processes and attempt to identify similar areas to yours where they are having problems and where they are satisfied with results.

FINAL CONFIGURATION AND SELECTION

Since each manufacturer or service firm will propose various alternative equipment configurations that respond to your specifications with varying degrees of performance and that may or may not meet your budget constraint, the final step should be to select the "best" overall configuration. With the budget limit externally set, the final choice will hinge on an analysis of features such as ability to handle the workload, systems growth capacity to handle future projected needs, reliability of the hardware and support services, existence of appropriate software, and obviously the total suitability of the proposed system to handle the specifications set down. Again, the utilization of technical consultants is highly recommended at this stage.

Hopefully, at this point non-responsive system configurations are obvious and can be quickly dispensed with leaving possibly only two or three to be considered. Among these choices the final decision will be much more difficult. In all likelihood no one of these final configurations will be totally superior to the others in all areas. The decision in the end may very will become a subjective choice based on such intangible factors as previous experience with the proposed firm, proximity of field offices, and cooperativeness of its personnel. It is also important to remember that while the system initially chosen may be compromise between cost and service that it should meet the overall strategic planning requirements identified in the preliminary and detailed study.

EPILOGUE

Selection of the final systems configuration does not terminate the responsibility of the detailed study. Installation and operation of the system cannot ever accomplish that end. The fact of the matter is that the need for the detailed study never ends. Once a system is operational, the study should be continued to periodically audit the progress and application of the system to the specifications drawn up and to continually identify new potential applications. While the intensity of the original studies will abate, the continued study of the functional areas will prevent any sudden shocks to the systems design and allow for gradual evolution into future equipment configurations and designs.

REFERENCES

1. Deckard, Noble S., "Capturing the Development Costs of an Integrated MIS," Journal of Systems Management (January, 1970).
2. Fried, Ray N., "Get the Computer System You Want," Harvard Business Review, (November-December, 1969).
3. Gross, Paul and Smith, Robert D. Systems Analysis and Design for Management, (New York: Dun-Donnelly Publishing Corp., 1976, pp. 207-245, 415-433.
4. Hare, Van Court, Systems Analysis: A Diagnostic Approach, (New York: Harcourt, Brace, and World, Inc., 1967, pp. 21-22.
5. Holmes, Robert W., "Twelve Areas to Investigate for Better MIS," Financial Executive, (July 1970).
6. Kranzberg, Melvin, "Computers: New Values for Society," The New York Times (January 9, 1967).
7. Massey, Daniel L. and Heptorstall, John, EDP Feasibility Analysis, (Braintree, Mass.: D. H. Mark Publishing Company, 1968).
8. Murdick, Robert G. and Ross, Joel E., Information Systems for Modern Management, (Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1971), pp. 473-504.
9. Ravity, Lester, "How to Approach A Quality Information System," Quality Progress (June 1969), pp. 22-23.
10. Rosen, R. J., "Computer Conversion: Getting Off on the Right Foot," The Canadian Chartered Accountant (June 1967).

A MANAGEMENT INFORMATION SYSTEM FOR ACADEMIC ADMINISTRATORS

Michael J. Powers

Applied Computer Science
Illinois State University
Normal, Illinois

Abstract

In times of decreasing financial support, declines and shifts in enrollment, and ever increasing demands for accountability the academic administrator has a critically important role to play as a manager of resources. In general the academic administrator, especially at the dean or department chair level, is not well prepared for this role; his appointment is more likely to be based on scholarly achievement than on management experience or skill. The problem is compounded by the general lack of information support systems tailored to the specific needs of the academic administrator. This paper examines the problem of providing such information support systems.

I. THE ACADEMIC ADMINISTRATOR AS MANAGER

The term "academic administrator" refers here to the positions of provost, dean and department chairperson. This is distinguished from the term non-academic administrator which includes the various business functions, admissions, registration and so on. The non-academic administrator at a university has long been recognized as a manager. Indeed, his duties and responsibilities are not significantly different from what one would expect to find in the commercial world.

The academic administrator, on the other hand, has not been traditionally viewed as a mahager. His appointment to and retention in his administrative position is more likely to be based on scholarly achievement than on management experience or skill. Whether good or bad, this is a reality based on a long tradition and hence is not likely to change appreciably in the near future. Unfortunately however, life is changing for the academic administrator in ways he cannot control. He may have been chosen as a scholar and he may be evaluated as a scholar, but his financial support is decreasing and student enrollments are declining and shifting around him. His very survival and the survival of those under him will be critically influenced by his ability to manage the resources at his disposal.

The problem, then, is that the academic administrator will have thrust upon him a very definite management role; but because of background and interest, he may be ill prepared to accept that role. It is unrealistic to believe that the problem can be solved by an overnight change in the reward structure under which he operates. The only feasible approach is (1) to train the academic administrator in how to manage (an especially critical need at the department chairperson level) and (2) to provide the information tools required to support his management function. This paper concerns a pilot project to define and provide the key information tools which can be used by the academic administrator to support his efforts in the management and allocation of resources. But it must be viewed in context. Reports don't make decisions, people do. Reports can support the decision making process, but their level of sophistication must match the level of managerial sophistication of the academic administrator who is using them. The approach, then, has contained the following elements:

- involvement of key adacemic administrators with at least a moderate level of "managerial sophistication in the initial definition of information requirements,
- provision of an initial set of tools to be used by a larger group of administrators within the context of "gently training" them in their roles as managers,
- revision and extension of the initial set of tools based on experience gained from their use,
- continued repetition of these last two steps.

The following section discusses the problems inherent in providing information support to academic managers. It is followed by sections which describe the multiphase approach used in the pilot project.

II. INFORMATION SUPPORT FOR THE ACADEMIC ADMINISTRATOR

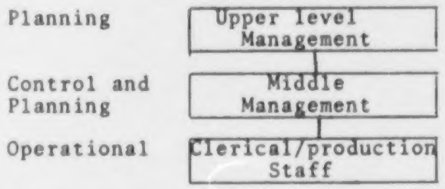
There are two basic problems inherent in attempting to provide information support for academic administrators. An understanding of the nature of these problems can help avoid many hours of frustration. To understand the problems it helps to first have a general understanding of the functions within an organization which require information support. A nice overview of these ideas, together with additional references is given in a paper by Foster and Gamble of Cincom Systems (see [1]). The point is to build a logical model of the organization based on what the organization does, not how it does it. The basic building blocks are events which are made up of tasks or activities. There are three types of events.

- Operational events occur on a day to day basis as the organization provides its service. Universities register students, schedule classes, assign classes to classrooms, hire faculty, provide student financial aid and so on. These events provide basic data about the university.
- Control events occur to monitor and regulate the operational events in order to optimize the performance of the organization within the context of objectives established by the organization. Data generated by operational events are used by control events. Control events in a university include monitoring class size and setting class enrollment levels, evaluation of faculty performance, budget control, and so on.
- Planning events occur in order to define objectives and to establish strategies for accomplishing these objectives. Again data generated by operational events are used, together with certain data from outside the organization (eg, national or regional college enrollment trends) to produce historical summaries and future trend information for use in the planning events.

The first problem is that information to support the planning events of academic administrators usually does not exist, at least in a format which is convenient and useable by a person who is likely to be (as seen in the first section) a novice manager. Most moderate to large academic organizations have well defined computer based systems to support the operational events. Examples include systems for class scheduling, registration, financial aid, the usual accounting functions, and so on. Control events are usually supported reasonably well by these same systems. Ideally, of course, systems to support planning events should also be based on these operational support systems. But in most cases systems support for planning functions is grossly deficient. The support which does exist is often a poor adaptation of a system designed to support control functions. Even that is usually directed toward the non-academic areas of the university, the governing board, and various external agencies rather than toward the academic administrator.

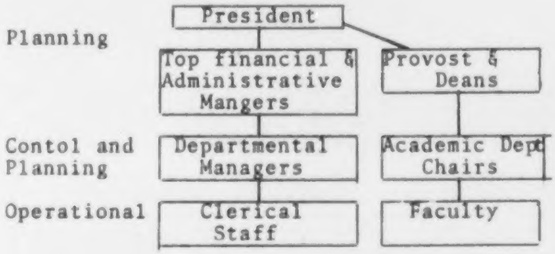
The second problem is that the organizational structure of the university can almost guarantee that information support for the academic administrator will not come easily. The placement of the computer and systems support function within the organizational structure of the typical university severely hampers the adademic administrator who wishes to define the information systems necessary to support his planning activity. In the typical business organization there is a direct vertical relationship from the clerical or production level, which is involved with operational events, through middle management levels, involved with control events, to upper level management, involved with planning events.

Business Organization



Upper management's planning activity is a direct outgrowth of the control and operational activities below it. Planning systems can be defined as natural extensions of the more basic systems. Equally important, upper management is in a position, politically, to ensure support for these systems. A similar vertical relationship exists in the academic institution, but it is based on the business/administrative side of the organization

University



78

instead of the academic side. Academic administrators are not part of this vertical structure. Their planning support needs do not grow quite as naturally out of existing operational and control systems. But more importantly they are not in a position politically to ensure the support they require.

Recognizing these problems helps the organization to deal with them. A significant amount of data concerning the operation of the university may well be available. But if it is not packaged as part of a system to support the planning activity of academic administrators, attempting to use it in that way will only produce frustration. Dealing with the second problem involves the solution of certain political problems; this will likely be both difficult and time consuming. Ultimately a solution is necessary; short term it may not be. The pilot project described here is being carried on totally outside the Computer Services area using the resources of the Applied Computer faculty and students.

III. DEFINITION OF THE BASIC SYSTEM

A. The context

The overall objective of the pilot project described here was to provide a set of information tools to support academic administrators in their managerial decision making processes. These tools would be directed almost exclusively to planning activities as opposed to control activites. While the actual design and implementation of such a management information system can be a significant challenge, the definition of requirements which must precede it can be even more difficult. The users of the system must, in the final analysis, make these definitions. Recognizing that the administrators involved had differing levels of sophistication in the use of computer reports as management tools, a multiphase approach was taken. The basic idea was to implement a first approximation to a solution as quickly as possible. It was an accepted part of the overall plan that this initial implementation was in part a tool by which user sophistication could increase. Suggestions for refinements would be expected as the tools were used, thus laying a solid groundwork for a more sophisticated set of tools in the future.

The first phase of the project centered in the College of Arts and Sciences with fifteen academic departments and eighteen distinct cost centers. The principal driving force behind the initiation of the project was the dean of the college who felt a strong need for reasonably rapid access to information which could support a plan for continuous evaluation of programs in terms of quality, trends, and efficient utilization of resources.

B. The basic reports

The first phase of the project was primarily concerned with the analysis of historical data, with an elementary forecasting capability to be added in phase two. The first step was to define a set of reports which could serve as a good first approximation to an ultimate solution and which were feasible to implement using limited resources and drawing off data produced by existing systems supporting the basic operational events of the university.

It soon became apparent that, if anything, the academic administrator was bombarded with too much data. He received too much detail in some areas, not enough in others, and rarely was it packaged in a format which supported his planning activity. The primary objective, then, was to define a department based report which would contain, in compact form, the essential information required to support basic planning for resource utilization at the department and college level.

The result was a very compact and readable Department Profile Report which presents in highly summarized form the key information, from a performance and resource utilization perspective, concerning a department's operation. The report for each department is less than five pages long and has columns to display up to nine years of information. College and university level summaries are also available in the same format. The information is organized in six sections outlined below.

1. Student Data
   - .Average number of majors and minors per semester categorized by undergraduates and graduates both on-campus and extension.
   - .Number of degrees awarded in that major for the year categorized by degree type.
   - .Placement of graduates for the year categorized by degree type and type of placement (teaching, non-teaching, advanced education, other).
2. Department Course Data
   - .Average number of students enrolled in courses within the department per semester categorized by course level.
   - .Average number of credit hours produced by the department per semester categorized by course level.
3. Faculty Status Data
   - .Faculty FTE allocated to the department categorized by regular lines and temporary lines.
   - .Number of faculty holding rank in the department categorized by

tenured, probationary, temporary and by percent of assignment with in the department.

- .The tenure status of the faculty, showing the number of tenured and probationary faculty by rank and the number of tenure decisions to be made that year.

4. Faculty Assignment Data
   - .The number of faculty FTE allocated and the number actually assigned in several categories: direct instruction, indirect instruction, administration, sabbatical, research, and other.
   - .The number of graduate and undergraduate assistants allocated.
   - .A teaching load ratio of credit hours produced to FTE assigned to direct instruction.
5. Faculty Activity Data
   - .The number of publications produced by faculty in the department in various categories.
   - .The number of research and service projects in progress.
   - .A measure of the external grants applied for and received.
6. Financial Data
   - .Dollars budgeted and actually spent for the fiscal year within each account category.
   - .A productivity ratio of total dollars spent to credit hours generated.

The tremendous power of this tool lies in its compactness and its completeness. The academic administrator has contained within five pages a complete profile of his operation for up to nine years. This report may well flag certain trouble spots which require a more detailed examination. With this in mind several slightly more detailed reports were designed. Two of these are briefly described below.

The Department Majors report shows, for each department, where its majors are generating credit hours. There is a two page report for each department, one for on-campus and one for off-campus majors. For each course level (100,200, etc.) the number of hours generated by the department's majors is shown. These hours are broken into five groups. The top four departments in which the majors generated hours that term are listed separately, with all remaining hours placed in the fifth group. The format permits up to five columns for which the user may specify the semesters desired. Trends in credit hour generation by a given department's majors are easy to spot.

The Course Enrollment Analysis report reverses the view given by the Department Majors report. It analyses each course offered by the department in terms of which majors are taking the course. For each course the number of enrollments are broken into five groups. The top four majors enrolled in that course are listed separately, with all remaining enrollments placed in the fifth group. The percentage of students successfully completing the course is also shown for each group. The format permits up to three columns for which the user may specify the semesters desired. Trends in student enrollments and the effects of shifting distributions among majors at the university are highlighted nicely.

The data base designed to support this system permits several other detailed reports; but the reports presented above are the key tools to be provided by the first phase of the development. While these reports cover only past history, they do provide the academic administrator with a helpful perspective on where his unit is coming from, together with any apparent trends. And this perspective is presented in a unified manner within the context of the entire operation of the unit, not as a random smattering of disjoint pieces of data. To be sure, the system is not sufficiently flexible and it lacks some highly desirable modeling capabilities. But it represents an effective first step. It is a useful tool in its own right. And it can be used to help academic administrators gain the necessary managerial sophistication to permit them to play an active role in the definition of a more flexible system.

C. An overview of the initial system design.

A detailed description of the system design is not approporate here. The design is too dependent on local conditions, especially on the local systems used to support the various "operational events". There are, however, several ideas which are transportable and hence of general interest.

This first phase of the system contains two major subsystems: a file building and maintenance subsystem and a report writing subsystem. The file building activity was designed to draw on existing detail files whenever possible, using separate manual input only in those instances when the required data was not already captured in machine readable form. The subsystem maintains four highly summarized sequential files.

1. The "program-majors file" contains one record for each major in each year/term. Each record contains the number of majors and minors, the number of graduates and their placement, and the hours generated by those majors in various departments (at the same level of detail as that used in the Department Majors report).
2. The "department-course file"

contains one record for each course in each department in each year/term. Each record contains enrollment data at the same level of detail as the course analysis report.

3. The "personnel file" contains one record for each department in each year/term. Each record contains the data to support the faculty status, assignment, and activity sections of the department profile report.
4. The "financial file" contains one record for each department in each fiscal year. Each record contains the budget and actual figures in each account category.

Typically each file is updated by summarizing several highly detailed files designed to support various existing operational event oriented systems. The procedure is clumsy, but need not be frequently performed. The clumsiness and inflexibility is unavoidable, due largely to the fact that existing "operational event" systems were not designed to accommodate "planning event" systems for academic administrators.

With respect to the report writing subsystem note that the implementation of the Department Profile report is not at all a straightforward problem. For each department, up to nine years worth of data from each of the four files must be used to print a single section of the report. Good structured design techniques are essential here.

IV. FUTURE DEVELOPMENT OF THE SYSTEM.

A forecasting capability will be added during the second phase of the project. Just as in phase one, the real challenge will lie in defining the capabilities desired and the rules under which they will operate. During the next year while the phase one reports are used by the Dean of Arts and Sciences and his department chairpersons as part of a review-evaluation-planning activity, a small subset of these people will begin to define the forecasting requirements. The approach will be simple and straightforward, providing only those capabilities which current academic administrators will be able to effectively use as management tools. Implementation will probably consist of a new subsystem to generate "future records" based on a set of input parameters and historical records in each of the four files of the data base. The input parameters will involve, among other things, assumptions about overall enrollment trends, shifting patterns of student interest, and financial support available. The report writing subsystem can then be used to display combinations of historical data and forecasted data in the familiar format of the reports from phase one.

The system, even at the end of phase two, will be far from ideal. It will be clumsy, only moderately responsive, and rather inflexible in terms of its ability to meet currently undefined needs. A real purpose will have been served; a set of tools will have been provided where none currently exist. Yet the more successful these tools are, the sooner they will become inadaquate. As the users become more sophisticated in the use of these reports as management tools, they will demand more timeliness and flexibility. It seems, at this point, that the ultimate solution must involve a redefinition of certain key "operational event" systems and their implementation in a database environment. This will be a major undertaking. It involves far more than plunging ahead and purchasing a database management system and then converting existing systems to it. It means first modeling a significant part of the organization in terms of operational, control, and planning events. Both the academic and non-academic arms of the organization must be involved. The fundamental thesis of this approach (see [1] is that the operational events of an organization remain fairly constant, while the planning events are the most volatile. To first model the organization and then establish the database design giving heaviest (but not exclusive) weight to the design implications imposed by the operational events will result in a more stable and flexible database design. Ideally phase three will include this organizational modeling and database software support in the key operational areas. This will provide the necessary technical base to support planning oriented systems for academic administrators. As a result of the first two phases academic administrators will have grown in their ability to use information systems to manage their areas, and they will have a clear understanding of their information requirements.

Reference

[1] Foster, Cory and Ocie Gamble, "Information System Architecture", Cincom Systems, Inc., Cincinnati, 1976.

DATA SECURITY & PRIVACY:
IMPLICATIONS FOR EDUCATION

S. M. RAUCHER
IBM CORPORATION

As computers in educational institutions take on increasingly responsible work, the more important it is to protect it from failure and catastrophe, and from criminals, vandals, incompetents, and others who would misuse its power. One assurance must be demanded: The Data processing function shall not lose vital data, introduce errors into them, or permit data to be read or modified without authorization. The subject paper will explore, in detail, the PRIVACY issue as it relates to educational institutions and the DATA SECURITY issues required to comply with the legal and moral obligation of today's educational data processing installation.

Without security of data from intentional or accidential disclosure or modification, privacy of information could not be assured. Let us not, however, take security and privacy as being synonmous. Privacy is a legal and morale issue. Data Security is an implication and management philosophy that may lead to further assurance of privacy.

Privacy means the legal and moral rights of individuals and organizations to determine for themselves when, how, and to what extent information about them is to be used and/or transmitted to others. Thus, privacy deals with information about me, and even though many people may be authorized to use that data about me, I want to control the form that they use that data. In effect, one might define privacy as the prevention of authorized people from making unauthorized use of data which they were authorized to collect.

Security may be defined as the protection of data from hazards, natural, or man-made, which may cause its destruction, modification, or disclosure, either accidentally or intentionally.

In 1973, the Department of Health, Education, & Welfare produced a report entitled, Records, Computers, and the Rights of Citizens. This report set forth a proposed code of fair information practice. Included in this code were the following points:

1. There must be no data recording system whose very existence is secret.
2. There must be a way for an individual to find out what information about him is in a record and how it is used.
3. There must be a way for an individual to prevent information about him, obtained for one purpose, from being used or made available for other purposes without his expressed consent.
4. There must be a way for an individual to correct or amend a record of identifiable information about him.
5. Any organization creating, maintaining, using or disseminating records of identifiable personal data must assure the reliability of the data for their intended use and must take reasonable precautions to prevent misuse of said data.

At the federal level, there are several statutes providing for specific privacy guarantees. One of these, The Privacy Act of 1974, is designed to provide safeguards for the individual against potential invasion of privacy by federal agencies. This law defines the rights of the individual to determine the contents of records, and their purpose, held about that individual by a federal agency. The individual has the right to prevent records from being used for other than their intended purpose and to gain access to those records, obtain copies and correct or amend those records. The Act also spells out the requirements on the federal agency to use identifiable personal data in a lawful and necessary manner and to require specific authority based on national security or other statutory authority in order to gain any deviation from the provisions of the Act. Of special interest in this legislation are the accountability provisions, which subject individuals as well as agencies of the government to civil suit for damanges occuring as a result of violation of the provisions of the Act. For example, an employee of a federal agency who fails to properly observe data security may be fined $5,000 per occurrence of the violation.

The relationship of the Privacy Act of 1974, to educational institutions is an indirect one. There are relationships for those institutions either working on governmental contracts; i.e., as contractors of federal agencies and therefore subject to the Act, and potential relationships for those educational agencies accepting federal grants.

Of more direct relation to educational institutions is the so-called Buckley Amendment to the Privacy Act; i.e., The Family Rights & Privacy Act of 1974. This legislation details privacy requirements for educational institutions. For example, the educational institution must permit the parent and/or student to review or receive copies of all educational records. Once the student reaches the age of majority (18 years), the parent's right to review student data is limited. (Internal Revenue Service rulings have concluded that parents maintain the right to review a student's record if the student is a dependent per IRS Regulations.) In addition, the Buckley Amendment provides that the educational institution must establish procedures to explain and interpret information and records, provide procedures whereby students and/or parents may amend incorrect data or receive hearings to determine whether such amendments are necessary. The educational institutions must obtain the specific consent of parent or student before disclosing information in its files with the exception of directory information, information necessary to other officials within the school system, information required to make financial aid decisions, or information required by state or other governmental agencies as directed by law. With the exception of these types of disclosure, the educational institution must maintain a detailed record of who has seen the information, who has made any amendments to the data, what was the nature of that amendment and who had authorized same. In addition, the parent or student may specifically request that their data not be included in those directory information areas alluded to above, and this request must be honored.

There are those who feel that the provisions of the Privacy Act of 1974, and the Family Rights & Privacy Act of 1974, are insufficient. As a result, a joint commission was impowered to study further statutory requirements potentially necessary to strengthen these areas. This Privacy Commission reported to President Carter in July of 1977. The report contained two sections of specific interest to educational institutions, implications for federally-sponsored research, and proposed amendments to the Buckley Amendment.

The Privacy Commission Report of 1977 suggests that federally-sponsored research should have limitations placed upon actions that might be taken as a result of accumulating personally identifiable data. For example, research data could not be subsequently used for prosecution of specific individuals. The legal requirements necessary in the collection and maintenance of this data should be restricted to audit purposes of the research project. Secondly, individually identifiable data should be maintained only as long as necessary to fulfill the purpose of the research project for which it was collected.

The Privacy Commission Report of 1977 proposed several areas of potential amendments to the Buckley Amendment. These include the establishment of procedures for reconsideration of administrative decisions that were based upon incorrect, inaccurate, or misleading data. Also, the elimination of the right of a student to waive his rights to inspect letters of recommendation or other parts of student files. The Commission also suggests the extension of the provisions of the Buckley Amendment to other agencies who serve quasi-educational purpose; i.e., testing or data assembly services that are under contract to the educational institutions. Applicants for admission to an institution should have the right to inspect their admissions records. Finally, the Commission recommends that sanctions for violation of the provisions of the Privacy Act should be made available in intermediate steps; i.e., short of full cut-off of federal funding.

While the work of the Privacy Commission continued, Congressman Barry Goldwater, Jr. (R-Calif.) and Edward Koch (D-New York) formulated a Comprehensive Right to Privacy Bill, which they choose to number H.R. 1984. The purpose of this bill was "to protect the constitutional right of privacy of individuals concerning whom identifiable information is recorded by enacting principles of information practices." While this legislation is currently pending before Congress, it is significant to note that the principles of H.R. 1984 are in concert with the suggestions of both the original and second Privacy Commissions and extend these principles to all parts of the public and private sector.

The administrative requirements of H.R. 1984 include "any organization maintaining an information system that includes personal information shall establish appropriate safeguards to secure the system from any reasonably foreseeable threats to its security." Stated another way, if enacted into law, this bill will mandate data security and provide penalties for not having reasonably prudent data security measures.

If the working definition of data security, is taken from the proposed federal legislation, we must begin a discussion of data security with a critical examination of the potential hazards which might befall data. Generically, the hazards to data fall in the categories of disclosure, modification or destruction by accidential or intentional means. The following lists potential hazards in a sequence based on the number of known incidents of the occurrence of the hazards.

1. Errors and Omissions
2. Dishonest Employees
3. Fire
4. Disgruntled Employees
5. Water Damage
6. Other

The most frequent hazard to data or programs is that of accidental disclosure, modification or destruction. Accidental disclosure might be caused by an operational error, the distribution of the report to the incorrect department or individual, or an error within the program itself that provides more information in the report than was authorized. The effect would be generally minimal unless a dishonest person were involved in obtaining the results and then making dishonest use of them.

Accidential modification to data may also occur as an operational error, such as mounting the wrong update tape, an input error or a programming error itself. The effect again may be minimal, if the loss is detected early. However, should such loss of integrity of the data perpetuate itself over a period of time, the cumulative effect may be significant. Destruction of data can similarly occur through operational errors such as dropping the disk pack thus not allowing the machine to be able to correctly read the information thereon. The maximum loss in this category would be the cost to recover by use of the back-up system.

Intentional modification or disclosure of data is typically the work of an individual who is attempting to be dishonest. Typically, the intentional modification of data can be prevented by never allowing the system to have a user work without "leaving foot prints in the sand," and publicizing this fact.

Similar hazards to those enumerated above for data must be considered in the protection of the programs that process the data. Should a program be accidentally modified incorrectly by a well-intentioned programmer, the effect may similarly be severe. Minimal effect will be achieved if programming errors or modifications are detected early. As was the case with the destruction of data, the destruction of programs, be it accidential or deliberate, may have the effect of the cost of backing up to a spare copy of the program or the cost to reprogram that system.

The second highest category of hazards to data security are by dishonest employees of the organization. This is particularly sensitive because 99% of the employees are honest and the other 1% are honest 99% of the time. Interestingly enough, security people tell us that most of the theft is not taking place in the computer room but in areas surrounding the information system. In fact, it is taking place in the main line of the activity whatever that happens to be. In a bank, the losses are taking place in the form of cash and securities, with an automobile manufacturer, it's in the form of parts stolen from inventory. Dishonest employees act rationally and attempt to minimize the possibility of being detected; that is, they attack in the place where the benefit versus risk of exposure is most attractive. Techniques such as providing increased accountability for action taken within the system and the ability to trace back who initiated various transactions tend to minimize the exposure from dishonest employees.

The next highest incident of hazards to data comes from fire. The greatest incidents of fire occurs in those areas surrounding the computer containing supplies like cards and paper and forms. These are highly combustible and are frequently stored within easy access to the computer room. The computer room is significantly less subject to fire than the office space surrounding it. However, the mission capability of the data processing system is none the less compromised by fire in these areas.

Next in sequence of hazards are the actions of the disgruntled employees which are frequently difficult to distinguish from errors, accidents, and omissions. This makes safeguards against disgruntled employees extremely difficult to guarantee. It places us in a position where we must minimize risk by providing minimal access to everyone.

Water damage, although lower on our incidents list is the second highest cost item in the category of hazards to data. Water damage also includes damage that is done by fireman and sprinkler systems, damages which were not listed under fire. Other types of water damage may also occur. Pipes may burst, drains may back up, storms may seep into the building, and floods may occur. Those installations with data processing equipment located in the basement or lower floors, will sustain the greatest potential risk of water damage from leaks occuring in the floors above or below.

In summary then, we can look at the potential hazards to data as being most frequently caused by people working within our institution who accidentally destroy the mission capability of the data processing installation.

As in all phases of the operation of an educational institution, data security risks may be minimized by adequate planning by management. An Emergency Plan should be established to limit the damage to the critical resources and protect the mission capabiility. This includes clear assignments of responsibility to communicate the alarm, evacuation and shelter plans, equipment protection plans, and a well defined division of responsibility amongst management and operational people to assist in the protection of resources in priority sequence.

Should an emergency occur, a Back-up Plan would then go into operation. The purpose of such a plan would be to allow the accomplishment of the highest priority sections of the mission between the time of the loss of mission capability and the time that recovery would be totally possible. The back-up plan must identify the critical mass of equipment, forms, and other activities necessary to provide the essential mission back up. The plan should identify where these activities may take place while your equipment is unavailable. (One must remember to provide the same level of security measures in the back up situation as are present in the normal operational mode.)

A Recovery Plan must also be in place to provide the capability to recover the full mission capability once lost. Again, alternate sites and sources must be available, back up copies of programs, systems, and data must be available. Communications facilities replacement must be considered as well as supplies and forms.

A Vital Records Plan should be established. This plan would identify those pieces of data that are essential to the well being of the employees, faculty, students, parents, and other key groups served by the institution. Off-site storage of back up copies of such records, protected storage in vaults on site, and the guarantee that off-site records are updated periodically, are all part of the vital records plan.

The Access Control Plan would limit the number of people who have physical access to a location to those people whose presence is absolutely mandatory to complete their assigned tasks. Access should be localized and under localized control. There should be multiple control points to assure that access is not breeched, and technology may be employed, such as badge readers, or control access systems, to both control access, note variances and to provide for follow up on breeches of security.

A Data Classification program should be put in place, such that all data is categorized based on its degree of confidentiality. All media should be labeled with the name of the set of protective measures appropriate to the data recorded on that media. This labeling might take the form of secret, confidential, sensitive, directory information, or other definitions the institution feels applicable to the level of sensitivity of that data. Once these precise definitions are known, individuals may be licensed to utilize data of given classifications. Key to a data classification program is management's commitment to discipline violation of the data classification profile.

With any of the items previous mentioned, it is essential that plans be periodically tested, updated, drills be performed, and all employees informed of management's commitment to data security.

There are several fundamental principles that may assist the institution in providing data security. First amongst these is the concept of minimal access. Sensitive duties should be performed by multiple people, each with minimal knowledge of what the other employee must do to complete the transaction. No one individual should have access to combinations of resources which when put together would provide a security level higher than that of the individual components. Facilities and procedures must be built into the system that would prohibit individuals from both accessing data and concealing the fact that data had been accessed. Similarly, good accounting practices dictate that no one individual should be able to both originate a transaction and approve the final results thereof.

An over-riding principle in providing data security is that of least priviledge. That is, risk is reduced by limiting access to resources. If the people who must interact with a system are restricted to the minimum set of resources they require to do their job, the risk that they may accidentally or intentionally breech security is reduced and the individuals themselves are protected from undue temptation and/ or suspicion of dishonest actions.

The system itself should provide certain basic security functions. These include identification of the user by some form of permanent identifier; e.g., employee number, authentication that the user who has just entered said identification number is in fact who he say he is; e.g., use of a password. A security profile should indicate the level of authorization of this user within the organization. Delegation of responsibilities by the security administrator of the institution should be clear cut. An audit trail should beprovided to indicate the condition of all data before and after various updates, who made the update, and under what authority. (Don't forget to provide the same level of security for journals and audit trail tapes as was defined for the initial data base). Finally, surveillance by the security administrator and management of the institution must be provided to control and monitor all actions within the system for compliance with the security provisions mandated.

Specific access conditions should be established to provide data security. This includes access to programs only through specified terminals. Access to data should also be via specified programs within the system and the data only available through specific keys or key values or part of a specific structural view of the data base. Access to resources including files, programs, machines, etc. should be within specified times.

In addition, access to resources may be further restricted to those individuals who have completed some specified training requirements and have thus exhibited the ability as well as authorization to utilize the system. Finally, access to resources may be necessary, on a short notice emergency basis and authorized through a well-defined approval mechanism. In any event, the assumption should be made that if not specifically authorized, the user is assumed to have no authorization. If not specifically authorized, the resource should be assumed to be unavailable.

Given the fact that 100% security is an impossibility, we must then define sufficient security to make the level of risk become acceptable. Let us define sufficient security as that level of security which yields minimal risks for an economically justifiable cost.

In order to develop the concept of acceptable risk, the instituion should develop a risk assessment program. This program would assure that all of management has adequate understanding of the exposures and risks of data security and that a strategy has been put in place to minimize the risk. A quantitative and qualitative assessment of the hazards, the potential damages should those hazards occur, and the resources affected by those hazards, should be made. The quantification of this risk assessment should then become a basis for a cost effective analysis of data security.

The choice is management's, to do something or do nothing. What to do is totally dependent upon both the significance to the institution of the resources being protected, the level of current data security problems within the institution, mandated requirements by Federal, State or Local statute, and other potential institutional problems. Management must make policy decisions as to who "owns" personal information and how much to spend on safeguards to provide data security. Risk assessment programs and security administration can set forth the detailed necessity for data security and the potential hazards thereto. Technology can provide certain levels of instrinsic data security both within the computer, the data base, and the surrounding physical facilities. But ultimately the choice is management's, and management must set the tone.

COMPUTER-ASSISTED INSTRUCTION

SUCCESS STORY:
CAI IN THE ELEMENTARY SCHOOL

Kathleen Adkins, Vice-President
Computer Curriculum Corporation

This paper establishes criteria for judging the success of CAI programs for elementary schools, describes results typical of successful programs, and lists some of the factors that are instrumental in creating a successful CAI program for elementary schools.

Computer-assisted instruction (CAI) is now a fact of life in many elementary schools. In examining the success of CAI programs in these schools, I shall address three questions: What results should we expect from a successful CAI program in an elementary school? What results have actually been obtained in typical elementary-school CAI programs? What factors contribute to the success of CAI programs in elementary schools?

WHAT RESULTS SHOULD WE EXPECT FROM A SUCCESSFUL CAI PROGRAM IN AN ELEMENTARY SCHOOL?

We should first formulate the results that determine a successful CAI program. I shall discuss results according to three criteria: academic gains, cost-effectiveness, and well-being of the user.

1. Academic gains. A successful CAI program should consistently produce significant gains in academic achievement as measured by standardized tests; these gains should be higher than those attained under ordinary classroom conditions.
2. Cost-effectiveness. The per-student cost of a successful CAI program should be reasonable; that is, the value per dollar spent should be as high as or higher than that of other programs which aim at producing the same results.
3. User well-being. A successful CAI program in an elementary school should not radically disrupt normal classroom procedures or be hard to use. It should make a positive contribution to the well-being of its users, both students and teachers, in ways that can be defined and measured.

WHAT RESULTS HAVE BEEN OBTAINED IN TYPICAL ELEMENTARY-SCHOOL CAI PROGRAMS?

Now that these criteria have been defined, let's examine their application to a typical elementary-school CAI program from Computer Curriculum Corporation (CCC). As far as we know, no schools have as yet analyzed their CAI program's effectiveness according to all three of the above-listed criteria; therefore I shall use data from several different schools to build a composite picture.

A typical elementary-school CAI program from CCC consists of a computer and up to 96 visual display terminals installed in schoolrooms designated as CAI labs. Although more curriculum is available, most schools use only CCC's elementary mathematics curriculum and one of CCC's elementary reading curriculums for grades 3-6.

Students spend ten minutes a day on each CAI course they are taking. They attend the lab for their ten-minute CAI session, then return to their classrooms. Printed reports on student or class performance are available at any time.

Academic gains. Most users of CCC's CAI programs are schools with large populations of Title I students. Students who participate in the CAI program are typically at least one year behind their school grade year in academic achievement as measured by standardized tests such as the SAT, MAT, or ITBS. Under ordinary classroom conditions, these students usually gain .7 years for every year they attend elementary school (Law, 1967). Over a period of several years, they fall further and further behind expected achievement levels. Any supplemental remedial program thus needs to raise achievement levels by more than one year per year of attendance in order to close the gap between actual and desired performance.

In order to be judged successful, therefore, a CAI program in a Title I elementary school should raise achievement levels by 1.1 years per school year or more. This outcome is desirable for non-Title I schools as well. In actual use, gains of more than one year per school year are not only attainable but predictable when the CAI program is used normally (normal use is defined as a 10-minute session for 150 days of the school year, or 1500 minutes of CAI yearly for each participating student). For example, in Drew, Mississippi, fourth through sixth grade students in a non-CAI supplemental program attained math gains of .6, .05, and .7 years on the SAT in 1975-76; in 1976-77 the same students gained 1.2, 1.2, and 1.7 years on the math section of the SAT using CCC's elementary CAI program. Gains on the reading section of the SAT were similarly dramatic, moving from .25 and .0 years' growth prior to the use of CAI in grades five and six to 1.5 and 1.2 years' growth after the introduction of CAI (Pettey, 1977).

In Shawnee Mission, Kansas, elementary students' math gains in 1976-77 with CAI from CCC ranged from 1.3 to 1.8 years as measured by the Kay Math Diagnostic Arithmetic Test in grades three through six. These gains occurred in spite of the fact that none of the grades' mean time spent on CAI came close to CCC's recommended 1500 minutes a year. Their 1976-77 reading gains with CAI, as measured by the Woodcock Reading Mastery Test, ranged from 1.5 to 2.1 years in grades three through six (Poulsen & Macken, 1978).

Of particular interest is the impact CAI can have on handicapped populations with low expected achievement rates. The Western Pennsylvania School for the Deaf in Mt. Airy, Pennsylvania, is composed of three smaller schools. Expected SAT gains for their students were calculated by averaging gains in preceding years; these gains were then compared to actual gains after use of CCC's CAI program in 1976-77. Actual math gains for the lower, middle, and upper schools respectively were 5, 5.7, and 2.9 times as high as expected gains. In reading, actual gains were 4.5, 3.9, and 2.6 times as high as expected gains (Holbrook, 1977).

Another handicapped group that has benefited from CAI is the educable mentally retarded. A group of 20 EMR students in Meridian, Mississippi, gained .99 years in math during only seven months of participation in CCC's CAI program in 1975-76 (The computer-assisted instruction end of year report, 1976).

These and other studies not cited here clearly show that a CAI program in elementary schools can meet exacting criteria for success in producing achievement gains.

Cost-effectiveness. Most of CCC's CAI lab configurations cost well under $100 per student per year; their yearly per-student cost is usually lower than the cost of adding aides to do an equivalent job. Their effectiveness per dollar invested is higher than that of other equivalent programs.

A report by Educational Assessment Service, Inc., (1976) for the City of Chicago illustrates this point. In the report, CCC's CAI program was compared with Prescription Learning and System 80. CAI students' gains were as good as or better than the gains of the other students, and CAI per-pupil costs were about one-third the per-pupil costs of the other two programs. In addition, CAI students spent about half as much time in the CAI program as the other students spent in theirs. From the points of view of time, cost, and gain, the CAI program delivered more value per dollar than the other programs.

Well-being of users. Many schools have reported attitudinal changes in CAI students that have resulted in decreased vandalism, less frequent behavior problems, and lowered absentee rates. Any of these changes improves the learning environment and contributes to the well-being of both students and staff.

Nelson D. Crandall reports that CCC's CAI program has reversed a 16-year decline in achievement gains in his Los Nietos, California, school district, a feat that several other supplemental programs, including one that reduced the student-adult ratio to 7 to 1, had been unable to accomplish. He speculates that the district's rapidly rising achievement scores are due partly to CAI's role in stimulating attitudinal change. According to Crandall, the CAI students have a sense of gaining internal control over events; that is, using the CAI program allows students to perceive that there is a cause-and-effect relationship between their efforts and their results. Crandall feels that this shift from external control (the feeling that events are the result of chance, fate, luck, or having influential friends) to internal control (the feeling that one's efforts have an impact and produce predictable results) is the most important attitudinal by-product of a good CAI system (Crandall, 1976). Studies currently under way are investigating whether CAI use does change the locus of control from external to internal.

WHAT FACTORS CONTRIBUTE TO THE SUCCESS OF CAI PROGRAMS IN ELEMENTARY SCHOOL?

Academic gains. CCC has found that consistently good academic gains arise from the use of carefully constructed CAI programs that employ the following strategies:

1. Individualization. Motion strategies that are flexible and responsive allow each student to work at an optimal balance between boredom and frustration.
2. Skill reinforcement. "Use it or lose it" applies as much to cognitive skills as to skills needed on the football field or at the piano. An effective CAI program provides abundant practice in the use of cognitive skills.

   To maximize skill retention, two factors should be taken into account: a. daily drill produces the best results; b. review of previously learned skills should be distributed over most or all of the academic year.
3. Immediate, positive feedback. Students need to know the result of each of their responses, and they need to receive feedback in less than two seconds.
4. Courseware that spans several grade years. A successful CAI program should have a grade range wide enough to accommodate many students at varying achievement levels and to allow any one student to receive continuous individualized practice, review, and introduction of new skills.
5. Abundance of courseware at each achievement level. The courseware should provide enough material at each level to give students intensive practice without repetition of individual items.

Lab management is important to the production of consistently high academic gains, too. CCC has repeatedly found for its elementary curriculums a direct correlation between amount of time spent working in the CAI lab and achievement gains as measured by standardized tests. Students who are scheduled for lab only once or twice a week, have high absentee rates, or for other reasons fall far short of attaining 1500 minutes of CAI per school year (10 minutes daily x 150 school days) consistently gain less than students who follow the recommended schedule. Students who attain close to 1500 minutes of CAI work per school year show uniformly high gains in achievement.

Cost-effectiveness. The key to keeping costs per student down lies in raising to a maximum the number of students who use the program. At the same time, the program must produce the desired results for the students who participate in it. To reach the optimal balance between number of students served and maximum effectiveness, CCC incorporates the following factors into its program:

1. Short session times. Short sessions allow more students to use the CAI program each day.
2. Courseware that addresses the most basic academic needs; that is, mathematics, reading, and language arts. The courseware is intended to serve the widest possible audience among its user population.
3. Continuing efforts to make available hardware adequate to the instructional task at prices competitive with those of other forms of supplemental instruction.
4. Courseware that produces the greatest possible effect in the shortest possible time.

User well-being. The following factors contribute to the well-being of the program's users:

1. Motion strategies are designed to place the student at a level where success can be experienced.
2. Mistakes essential to the learning process are made in a private arena, which allows the student with a history of poor achievement to give answers with a degree of psychological safety sometimes not available in a group of peers.
3. Immediate feedback allows students to connect their actions with specific results in the environment, establishing a causal relationship between action and result.
4. The program is easy to use and can be fitted into a school's normal procedures without massive upheaval.

CONCLUSION

A successful CAI program competes well against more traditional programs in both cost and effectiveness. The objective of delivering a highly productive CAI program whose cost is reasonable has pronounced implications for the design and construction of all aspects of the program. The hardware must be reliable and must not tie up funds in capacity unnecessary for effective courseware; the programming must allow a large number of students to be served without appreciable delay or degradation of response. The courseware should strike a balance between richness of response and avoidance of instructional trivialities. In order to serve large numbers of students with maximal effect, it must pare away unnecessary frills and avoid tempting sidetracks into activities which are exciting but less productive.

These strategies can and do work well, as CCC's many CAI installations in elementary schools demonstrate. Good CAI programs can make vital contributions at reasonable costs to students whose achievement levels are most in need of improvement. In addition, elementary-school CAI can increase the individual student's sense of self-worth and enhance the total environment of the school.

Computer assisted instruction end of year report, school year: 1975-1976, Meridian, Mississippi. Unpublished manuscript, 1976.

Crandall, N. CAI: Its role in the education of ethnic minorities. The Journal, December 1976, p. 24.

Educational Assessment Service, Inc. Comparative analysis of three ESEA Title I reading and mathematics activities. City of Chicago: Board of Education, 1976.

Holbrook, J. Analysis of Philadelphia school for the deaf, CAI results. Unpublished manuscript, 1977.

Law, A. S. The California State Department of Education 2nd annual report, Title I. State of California, 1967.

Pettey, G. F. Personal communication, October 19, 1977.

Poulsen, G., & Macken, E. Evaluation studies of CCC curriculums, 1975-1977. CCC Educational Studies, 1 (2), Palo Alto, California: Computer Curriculum Corporation, 1978.

English Usage Exercises:
A CAI Series

John G. Allee
The George Washington University
Peter J. Quinn
Robert L. Williams
St. John's University, New York

English Usage Exercises I (EUX) is a set of 96 computer terminal sessions that explore verb-pronoun relationships that often reflect written usage problems for many students. The intended audiences for EUX are the intermediate, middle, and junior high school levels. Each EUX session is composed of a short didactic lesson and a set of sentence exercises. Each session is designed for an average sitting of 10 to 12 minutes. The 96 Sessions, grouped in 21 series topics, follow a developmental and sequential format.

EUX may be run on either cathode ray tube or conventional terminals. EUX sessions are written in Dartmouth BASIC. Algorithmically written, each session program stands alone, the sentence exercises and their questions and comments being composed from data statements within the programs.

A session of EUX follows:

```
EUX093

                    D ENDINGS OR MARKERS III

TYPE YOUR FIRST NAME.  ? ROD

ROD, WE LEARNED THAT 'DO' OFTEN HELPS OTHER VERBS:

             OUR PAL OFTEN >DOES< ASSIST US.

'DOES' KEEPS PRESENT TIME.  HOW COULD IT BE CHANGED TO
KEEP PAST TIME?

             OUR PAL OFTEN >DID< ASSIST US.
     READY?

WHILE MANY VERBS LIKE 'TALK' OR 'CALL' KEEP PAST TIME WITH
'ED' ENDINGS, 'DO' IS ONE OF SEVERAL VERBS WHOSE SPELLING
IS ALMOST COMPLETELY CHANGED WHEN THE 'D' MARKER IS USED.

PRESENT:  MY FRIENDS NEVER DO CALL ME.
PAST:     MY FRIENDS NEVER DID CALL ME.

     READY?
LET'S PRACTICE WITH PRESENT AND PAST 'DO' VERBS.

EXAMPLE:  OUR PAL OFTEN ----- WALK WITH US.        PRES? DOES
        O. K.
EXAMPLE:  OUR PAL NEVER ----- HELP US.             PAST? DID
        O. K.

  1 . OUR SISTERS OFTEN ----- TALK TO US.          PRES? DO
        O. K.

  2 . HER PALS SELDOM ----- HELP  HER.             PAST? DO
        OOPS.   DID

  3 . THEIR BROTHER ALWAYS ----- WALK WITH THEM.  PRES?
        TRY AGAIN? DOES
        O. K.

  4 . MY CLASSMATE NEVER ----- CALL  ME.           PAST? DID
        O. K.

  5 . YOUR TEACHERS FREQUENTLY ----- YELL AT YOU. PRES? DO
        O. K.

  6 . HIS FRIENDS SOMETIMES ----- HUMOR  HIM.      PAST? DO
        OOPS.   DID

  7 . OUR SISTER RARELY ----- PLAY WITH US.        PRES? DID
        OOPS.   DOES

  8 . HER PAL USUALLY ----- GREET  HER.            PAST? DID
        O. K.

  9 . THEIR BROTHERS GENERALLY ----- TAUNT  THEM. PRES? DID
        OOPS.   DO

 10 . MY CLASSMATES OFTEN ----- PAMPER  ME.        PAST? DID
        O. K.

 11 . YOUR TEACHER SELDOM ----- TOIL WITH YOU.     PRES? DOES
        O. K.

 12 . HIS FRIEND ALWAYS ----- TALK TO HIM.         PAST? DID
        O. K.

 13 . OUR SISTERS NEVER ----- HELP  US.            PRES? DO
        O. K.

 14 . HER PALS FREQUENTLY ----- WALK WITH HER.     PAST? DID
        O. K.

 15 . THEIR BROTHER SOMETIMES ----- CALL  THEM.    PRES? DOES
        O. K.

 16 . MY CLASSMATE RARELY ----- YELL AT ME.        PAST? DID
        O. K.

 17 . YOUR TEACHERS USUALLY ----- HUMOR  YOU.      PRES? DO
        O. K.

 18 . HIS FRIENDS GENERALLY ----- PLAY WITH HIM.  PAST? DID
        O. K.

 19 . OUR SISTER OFTEN ----- GREET  US.            PRES? DOES
        O. K.

 20 . HER PAL SELDOM ----- TAUNT  HER.             PAST? DID
        O. K.

ROD, OF 20 QUESTIONS, YOU ANSWERED 16 CORRECTLY.
```

This is the third session of the ninth series entitled "D Endings or Markers," the 47th lesson of EUX. Most sessions are similar to this one. A short mini-lesson is presented, the student advancing the lesson by touching the "return" key. Then a set of sentence exercises, usually prefaced with examples, is presented. Most sessions ask the student to supply verb or pronoun forms to fill blank slots. Some lessons, however, require descriptive responses, the responses being words like "past" or symbols like "D" or "I," as exampled below:

```
NOW, LET'S PRACTICE.  ANSWER WITH THESE WORDS:
          PAST               PRESENT

    READY?

  1 . YESTERDAY WE WERE SMART.        'YESTERDAY' KEEPS ? PAST
       YES.

  2 . THEY ARE FUNNY TODAY.           'ARE' KEEPS       ? PRESENT
       YES.

  3 . YESTERDAY WE WERE HAPPY.        'YESTERDAY' KEEPS ? PAST
       YES.

TELL ME WHETHER THE SENTENCES I WRITE ARE 'D' OR 'I' TYPES.
'D' IS DECLARATIVE, OR A STATEMENT.  'I' IS INTERROGATIVE,
OR A QUESTION, ROD.

EXAMPLE: ARE WE AND MARIA HAPPY                  ? I
       YES.


  1 . HE IS SAD TODAY                            ? I
       NO.  IT IS DECLARATIVE.

  2 . MARIA AND HE WERE FUNNY THEN               ? D
       YES.

  3 . WAS HE SINGING YESTERDAY                   ? I
       YES.
```

The lessons essentially follow a sequential and developmental plan. Concepts developed in one series lead into the next where instructionally feasible. Once a topic has been treated in a series of lessons, it is left, the next series taking up a related topic. Later, after a number of additional topics have been touched upon, the first one is returned to for additional development or practice. In some instances a topic may be a review or may serve as a bridge between two more important ones.

The 21 topics of EUX and the number of sessions for each follow:

Counting with pronouns	9
Counting throughout Sentences	2
Double Subjects	4
Keeping Time	7
Ing Verbs	6
Object Pronouns	5
Questions	4
S & Zero Endings	7
D Endings or Markers	4
Verbs and Questions	3
Counting Review	5
This & That	5
Possession	5
It & They	3
Subject and Object Pronouns	6
Have	3
Have & Do	5
Helpers	3
Contractions--Not	4
Contractions--Verb	3
D Forms	5

The term "counting" is used simply to avoid a grammatical term--"number"--that may have proved difficult for some reluctant students. In a similar manner "double" has been substituted for "compound" and "keeping time" for "tense." In EUX "Ing," "S," "O," and "D" have been used as symbols for the verb forms traditionally referred to as present participle, present singular, present plural and past. These terms and symbols are used to help discuss language with the student: they are not the subject matter of EUX. EUX keeps its focus upon verb and pronoun usage concerns that often pose problems for many students.

EUX has been designed to assist the teacher help students with written usage problems. It was not conceived to help directly with oral usage problems. EUX is just one tool of many to be used with the variety of student usage backgrounds the teacher faces. For the student who accepts its challenge, EUX can help him discover and understand some of his usage problems. But obviously, to improve his writing, the student must write and be assessed by his teacher frequently.

Each session of EUX is a BASIC program. Each program stands and operates alone. The programs draw upon no files. The mini-lessons that begin the sessions reflect fixed print statements, which usually compose about one-third of the typical program. The sentence exercises, on the other hand, are generated from small arrays of words taken from the data lines of the programs. The data statements of the session exampled above follow:

```
90 PRINT YS;", OF";X;"QUESTIONS, YOU ANSWERED";X-S;"CORRECTLY."
91 DATA OFTEN,SELDOM,ALWAYS,NEVER,FREQUENTLY,SOMETIMES,RARELY
92 DATA USUALLY,GENERALLY,PRES,PAST,OUR,HER,THEIR,MY,YOUR,HIS
93 DATA US,HER,THEM,ME,YOU,HIM,SISTER,PAL,BROTHER,CLASSMATE
94 DATA TEACHER,FRIEND,LIBRARIAN,TALK,HELP,WALK,CALL,YELL,HUMOR
95 DATA PLAY,GREET,TAUNT,PAMPER,TOIL,"TO ","," ","WITH ","," "
96 DATA "AT ","," ","WITH ","," ","," ","," ","WITH "
97 END
```

Many sessions, or programs rather, are capable of generating considerably more exercise sentences than they have been set for. But careful observations of and discussions with students who have completed EUX have shown that 20 sentences compose a reasonable sitting.

A companion volume, A CAI Approach to the Development of English Skills: English Usages Exercises I, has been published, essentially being a guide developed for the classroom teacher. After a reasonably thorough discussion of the content, organization, and philosophy of EUX, each series of EUX is described by a set of notes and a brief outline, as exampled below:

9 D ENDINGS OR MARKERS EUX081-084 79

The series 'D Endings or Markers,' composed of four lessons, introduces past forms of action verbs with eD endings. Again the verb vocabulary from previous series is continued. Present tense 0 and S markers are continued in the exercises as contrastive counterparts to the D marker. Also, only noun subjects with pronoun determiners are used.

Lesson three introduces the past form of the auxiliary DO. And lesson four mixes action verbs with and without the DO auxiliary.

Time words are not used as signals in the sentences, as in earlier series. Instead the student is given the terms 'present' and 'past' and asked to supply the appropriate form without other clues except frequency.

My pal often > help < me.	Pres? helps
My pals often > help < me.	Past? helped

Outline

Lesson	Task	Focus	Sentence Patterns	Sentence Forms Decl	Sentence Forms Int
1	Spell present or past forms	help 0 help S help eD	Sub Verb Obj	x	
2	•	•	Sub Verb Obj Sub Verb P P	x	
3	•	help 0 help S help eD do 0 help 0 do eS help 0 do D help 0	•	x	
4	•	•	•	x	

Following each set of notes are partial runs of all of the sessions of the series. Each didactic lesson of a session is presented in full: each lesson is followed by some of the sentence exercises of the session. In effect, the volume gives the teacher and principal an easy to handle, comprehensive guide and manual for EUX I. EUX II, a volume of approximately 100 sessions that continues the topics and contents of EUX I, is in preparation.

FOUR APPLICATIONS RELATING TEACHER NEEDS TO A TIMESHARE SYSTEM

Developed by: MECC User Services
Presented by: John Arneson and
Robert Jamison

Four areas of study in the elementary/secondary surriculum will be examined from the view that needs of teachers generate the content and the approach of an application area. Need 1: Simplified instructions with built-in aids for the teacher of language arts to prepare individual drills and exercises "tailor-made" to the student's day-to-day, week-to-week skill development. Need 2: Background introduction, case studies and simulating a brief round of negotiation involving high school students so they can examine the process of collective bargaining. Need 3: Experiences for grade school students in the complex issues of business and economics, namely: pricing, advertising, competition, and overhead. Need 4: County Extension Services to tap computer resources of high schools for realistic data calculations and projections in machinery, cattle, and feed problems.

Curriculum on the elementary and secondary education level in Minnesota is constantly examined, reviewed, analyzed, and revised as the needs of students change. Individually and collectively students exhibit what they do not understand, what they want to learn, what seems important in the adult world, and what motivates them to achieve. Students challenge teachers to provide a curriculum that is relevant and to use methods of instruction that insure maximum learning.

After reviewing the areas of study in which a computer timesharing system can provide motivation and a simulation of real-world experiences in the curriculum - teachers have written the programs. Teachers have tested the programs and the mini-courses of study with groups of their own students. Then these teachers have shared these programs with other teachers through the Minnesota Educational Computing Consortium (MECC). Each program selected herein as an example of a mini-course of study is one that MECC supports with a manual which develops in detail the model for the computer program and explains some effective ways that the program can serve as the basis of a study in the classroom.

The four applications examined in this paper, and covered fully in documentation manuals, address a need that students and teachers recognized as valid "holes" in their curriculum. These needs and their "solutions" are as follows:

Need 1: Because students need various helps to build skills in language arts, teachers look for a means of preparing individual drills, exercises, and tests which emphasize only the skills requiring practice. Solution: A series of six programs covering 27 concepts have been developed with simplified instructions, built-in aids, and a similar mode of operation for each program, so teachers can prepare drills or worksheets for duplication or for single student use.

Need 2: Students of economics rarely get beyond the terms in collective bargaining to the level of negotiation. Solution: The BARGAIN program with accompanying documentation booklets provides case studies and a simulation wherein the student assumes the role of negotiator for the union or for management.

Need 3: Grade school students lack an experiential understanding of the issues in business and the adult world of economics. Solution: The SELL series of four programs introduces students to the implications of pricing, advertising, competition, and overhead.

Need 4: In agricultural communities of Minnesota the County Extension Services are the backbone of adult resources for the farmers. Their offices were not aware of the computer resources available to students in the high schools and vocational schools of the state.

Solution: Cooperative efforts between the County Extension Services, the University of Minnesota Agricultural School, and the Future Farmers in the high schools provided realistic data calculations and projections for programs modeled on machinery, cattle, and feed problems.

The remainder of this paper explores the computer programs designed to answer the above established needs. Models for representative programs will be explained, classroom applications and techniques will be identified, and an evaluation of the success of each mini-course will be presented. The paper will conclude with a brief discussion of the organization that makes possible the support of the computer in the classroom and its resources and personnel for facilitating widespread instructional computing.

Language Arts

Background: Mathematics and language arts are two areas of the elementary curriculum in which students must develop skills as a foundation for future learning. These skills are important in a pyramid of ascending abilities and each skill, while related to another skill, maintains a certain autonomy that makes it possible to isolate and develop it independently. Teachers in the elementary grades are required to be specialists in many areas of importance to the elementary child without neglecting any one area.

A teacher working with students identified as 'slow learners' recognized the need to provide a help to other teachers who lacked time to develop the drill exercises that could be used with a timeshare system. He saw that if a set of computer programs identified the skill areas of language arts that most teachers find need special practice with students, then this same set of programs could provide remediation for students whenever and wherever a computer terminal was available.

Through a Mini-Grant provided by MECC a set of programs was written, tested, and disseminated to all elementary teachers in the state. A year later the programs were expanded so that each behaved in a similar way. More skills were added to the list of those available and the programs became adaptable to all levels of the language arts curriculum, kindergarten through senior high school.

Model: The current language arts package consists of five programs covering word attack skills: READING, CONSONANTS, VOWELS, WORDS, and SENTENCES. Each program is divided into several lessons and each lesson concentrates on a specific word attack skill. For each skill at least one drill has been established with a word bank of 50 to 100 examples to generate a sample drill. (A concern in setting up the word banks was that teachers would not be able to agree on the words to be included, however, this has not been a problem and the word banks have been widely accepted). There are provisions within the programs for teachers to develop their own word banks.

A typical example of a drill a student might receive would be a lesson on the variations of consonant sounds. In the VARIANT lesson, there are three different drills available for use. (Other lesson areas have from one to five word banks established). To run a drill on the variant sounds of the letter 'C', a student accesses the program about consonants at the terminal. When the program requests the lesson name, the student types VARIANT. The drill is specified by typing the name CSOUND. Instructions are available if the student wishes them. The teacher may also decide to allow the student the choice of the number of exercises to try. The drill provides two chances for a correct response and when the exercise is finished the student is offered the option of reviewing the problems that were missed.

Applications/Evaluation: Teachers on the elementary and junior high level have found these programs with their 38 drills and word banks to be the tool they have been seeking. Use of the MECC Timeshare System is more extensive on the senior high level, but as elementary and junior high teachers experience the potential of the computer terminal as a remediation tool, they are requesting terminals and computer access in greater numbers. Teachers have the option in this set of programs of generating worksheets for a class of students, or a variety of worksheets on the same topic for groups of students. The programs give the teacher control over the number and variety of worksheets to be generated on each skill area. Future plans for this set of programs include the expansion of the package to comprehensively cover all reading skills.

Collective Bargaining

Background: Collective bargaining is the process whereby workers pool their power and negotiate with management as a unit, rather than each person negotiating individually. The study of collective bargaining consists of two topics: the negotiating process which makes it "bargaining", and the labor organizations that use bargaining to make it "collective." There is a definite need for the study of negotiations as everyone negotiates and many people are affected.

A teacher in Southern Minnesota wanted to include the study of collective bargaining in his senior economics class. Most textbooks limit the subject to a general introduction of terms. The teacher did some background research in college texts and newspapers and developed a unit that included case studies from the annals of Minnesota history. After testing, revising, and writing the computer simulation, the BARGAIN program was developed with support documentation for teacher and student. The unit of study encompassed the terminology of collective bargaining, but went on further to include experiential understanding of the process.

Model: When labor and management negotiate on the salary and fringe benefits to be paid for work, three options are available: mediation, arbitration, and a strike which represents a failure of negotiations, may disrupt a state's economy, and may affect the company and workers permanently.

In running BARGAIN, a simulation of the collective bargaining process, the student assumes the role of negotiator for the union or for management. An hourly wage is requested or offered by the students and the reaction of the other side is presented by the computer. The position of the two bargaining parties may then be represented by one of the following figures:

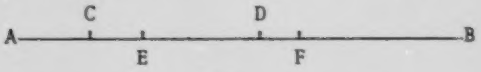


Figure 1

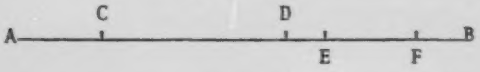


Figure 2

Consider line A-B as the range of possible agreements between the two parties. Line segment C-D represents the range of agreements that management would accept, while line segment E-F represents the range of agreements that labor would live with. Line segment D-E (if it exists, see Figure 1) is the common territory wherein a settlement can be found. The problem is that in many cases the two line segments (C-D and E-F) are disjoint (see Figure 2) and a strike takes place.

In the computer program the students do not know exactly which figure is the true representation of the two bargaining positions, and so must 'feel out' the other party without conceding more than is acceptable to the side represented.

Applications/Evaluation: The computer simulation with booklets for teacher background and student study are suggested for a unit in American history, American government, and economics at the secondary level, or grades 9-12. (The student manual provides three case studies from Minnesota labor history: Red Wing Pottery Strike of 1967, Albert Lea Packinhouse Strike of 1959-60, and the Minneapolis Truckers Strike of 1934. As the computer simulates the collective bargaining process, the students assume the role of negotiator, discuss the implications of each offer by the other side, and present a new 'offer'.

Teachers have found that simulations provide an experience that is unequaled for giving students an understanding and grasp of a collective bargaining situation. The case research studies in the documentation booklets is a great timesaver for Minnesota teachers; such timesaving is a major advantage of sharing materials within the state.

Business Application

Background: The need to provide elementary school students with experiences in business and economics was brought to the attention of a computer coordinator of a rural Minnesota school district by elementary teachers in the district who had received an intensive summer institute on economics and were developing units to teach basic concepts. Access to a computer was available to all elementary classes and the Huntington II simulation MARKET was used with ten sixth-grade classes. In the MARKET simulation two companies compete in an environment in which students must set the production levels, advertising budgets, and cost per unit for their companies.

Even though MARKET was designed for use with high school classes, results of this experiment with sixth-graders indicated that a computer simulation can be an effective method of providing students with the real-life experiences of the managers of a business. Furthermore, as students learn to make decisions and to live with the consequences of their decisions, they also learn the dynamics of a business.

However, a few problems with MARKET were isolated and discussed. Teachers felt that too many concepts - production, advertising, pricing, profit, competition, assets - were introduced to students too quickly and they were unable to adequately grasp the relationships between the concepts.

Model: A need was defined to develop a series of computer simulations to teach concepts in MARKET one at a time. The hypothetical business should involve costs and sales that students could comprehend. The structure of the computer program should encourage small groups to confer so all students can share in the decision-making. With the support of a MECC Mini-Grant the SELL series was developed.

In SELL Apples a student works exclusively with the concept of 'price'. Students are given an unlimited supply of apples to sell. Their task is to find the 'best price' to charge for their apples. Each day the students set the price for each apple, are told how many apples they sold, and what their income was for the day. By changing their price each day, the students are able to analyze how price affects the 'number sold' and 'income'.

SELL Plants builds on the student's knowledge of pricing and adds the concepts of 'advertising' and 'overhead'. On the first day students set the price to charge for the plants and the computer gives the number of plants they sold. On succeeding days the price remains 'fixed' and they are given the option of making advertising signs (at a fixed cost) to help increase sales. By first selling plants with no advertising, and then varying the number of signs each day, students investigate the effect that advertising has on 'income' and 'profit'.

SELL Lemonade provides the opportunity for two groups of students to make decisions on their business ventures without affecting each other's results. Random events built into the program forces them to make 'risk' decisions based on the previous day's experience. (See Figure 3).

SELL Bicycles adds the effects of 'competition' and uses numbers in the magnitude of a small business. Two bicycle companies compete directly with each other for a larger share of the market; decisions of one company directly affect the other.

Applications/Evaluation: SELL Apples has been used successfully with third grade students. In some Minnesota schools, the SELL series is used in grades 3, 4, 5, and 6, successively. SELL bicycles has also been used as a culminating program in an upper elementary economics unit with a full class of students. The mathematics needed to do the calculations performed by the computer are more challenging, providing a good experience for high school business and math classes. Student competition and the versatility of SELL Lemonade has made it the most popular program in the series. Not only elementary classes, but many junior and senior high math, social studies, and business classes have used SELL Lemonade successfully. Math teachers use the program by informing students of the number of glasses sold, but withhold other printout information until students calculate their own income, expenses, profit, and total assets. The option permitting a class to continue a game by entering the previous day's assets, allows this simulation to continue for a month or more.

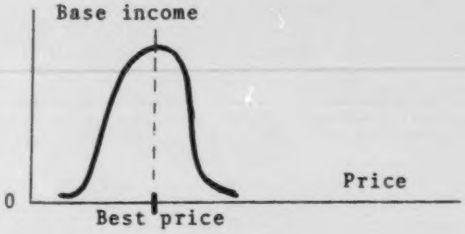


Figure 3

The model considers sales without advertising. Figure 4 shows how this model changes when advertising costs must be considered.

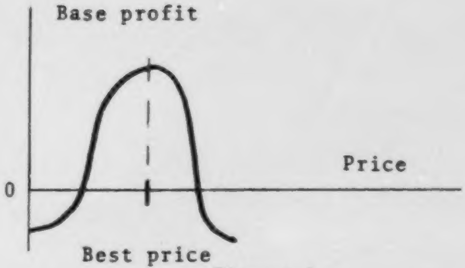


Figure 4

Agricultural Decision Aids

Background: In an ever-changing society the art of farming has become one of the most difficult businesses to manage. Today a farmer should be skilled in management and decision-making as well as in all the skills needed on a day-to-day basis. A chance to make decisions and make some mistakes using simulated data, is one of the most useful learning experiences for a student of agriculture. This concept became a reality through the Agricultural Extension Service of the University of Minnesota. A series of decision-aid programs were designed to reat such areas as amortizing a farm loan, buying and selling

Model: The package of agriculture decision-aids consists of 22 computer programs documented in a user guide. The model for each program varies. However, the models are consistent in the need for an input form that is best completed before coming to the computer terminal to run the data. When running the program, the student provides the information when and as requested. The output for each program not only supplies the calculations for the situation in the program (i.e., costs of feeding a certain ration to pigs, the expected crop returns on certain plantings, or the expected return on certain farm machinery), but also provides a printout of the breakeven points along with the profit and loss statistics.

Figure 5 is a partial printout from a run of the program FBEEF, a program that examines the current feed and feeder cattle prices along with expected future slaughter cattle prices. This is the summary of the results a student farmer could expect at other prices and also what price would be minimal to meet expenses for labor and facilities on the farm.

```
Return Per Head For Labor & Facilities with Different Prices

Selling       When Purchase Cost Per CWT Is:
Price/CWT     33.50     33.50     37.50       39.50      41.50

  37.25       10.38       .47     -9.45      -19.36     -29.28
  39.25       33.38     23.47     13.55        3.64      -6.28
  41.25       56.38     46.47     36.55   B   26.64      16.72
  43.25       79.38     69.47     59.55       49.64      39.72
  45.25      102.33     92.47     82.55       72.64      62.72

Break Even Selling Prices That Will Cover, Feed, Cperating
And $15.00 Return For Labor And Facilities:

Pu-chase      When Corn Price Per BU is:
Price/CWT      1.76      1.98      2.20        2.42       2.64

  33.50       34.84     36.25     37.65       39.06      40.46
  35.50       35.70     37.11     38.51       39.92      41.32
  37.50       36.57     37.97     39.33   C   40.78      42.19
  39.50       37.43     38.83     40.24       41.64      43.05
  41.50       38.29     39.70     41.10       42.51      43.91
```

Figure 5

Applications/Evaluation: These programs are used by students of agriculture at both secondary and post secondary institutions.The majority of the schools in Minnesota (outside the metropolitan area of St. Paul and Minneapolis) have an agriculture curriculum where these programs have met with immediate educational success. Farm management instructors who work with young farmers use these programs to test actual data to evaluate their decisions for long and short range planning. In some sections of Minnesota, the agriculture instructors have portable terminals that they use at the farm site. This makes it possible for students to make several runs of the program and look at all possible options, getting immediate results for discussion and change.

Conclusion

The computer-based instructional materials discussed in this paper are but a few of the activities supported and promoted by the Instructional Services Division of the Minnesota Educational Computing Consortium. MECC is a statewide service organization that provides instructional timesharing to all levels of education. This service includes access to the computer, a library of quality programs, technical expertise to assure a variety of educational features in programming, professional assistance to teachers by master teachers in creative ways of using the computer in the classroom, and support materials to continue the quality use of the computer as a tool in education.

The Instructional Services Division of MECC employs seven instructional coordinators who serve the elementary/secondary/vocational schools of the state, two coordinators who serve the higher education users of the state, a telephone consultant available to all users statewide, four managers and secretaries to provide direction and assistance in the overall planning and coordination efforts, plus a staff of technical programmers, computer operators, and telecommunications experts to functionally maintain the network from user to computer to user.

In the four year history of MECC usage has stabilized with 70 percent of the state's school districts using instructional timesharing and all higher education institutions identified as users of the computer in a variety of instructional ways. Programming and the library of programs comprise instructional computing. A structure that provides a vehicle for teachers and students to submit programs of their own initiative to other users in the state for evaluation has been available for two years. The improvement of present programs and the development of more applications is assured.

A COMPUTER-ASSISTED MATHEMATICS OF FINANCE COURSE

Jonathan C. Barron
Eastern College
St. Davids, Pa. 19087

ABSTRACT: This paper is an account of a computer-supplemented course in the mathematics of finance. It explains how topics from the mathematics of finance may be used to introduce various BASIC statements and elementary computing techniques. Included in the paper are several of the programs used in the course plus a discussion of its advantages and disadvantages.

INTRODUCTION: In the past decade, there has been a significant change in the role of the computer in undergraduate education. This change is a direct result of the development of time-sharing computer systems which have made the computer readily accessible to both students and faculty. This easy accessibility is possible because time-sharing enables many users to use the computer simultaneously by means of terminals which can be located almost anywhere.

As far as the user of a time-sharing system is concerned, there is no "turn around" time, the computer seems to respond instantaneously. Using the computer becomes a dialogue between the user and the system. Coupled with the fact that a computer terminal can be located in the classroom it is easy to see why the computer has become an effective classroom tool. The computer has moved into the classroom thereby acquiring a new role in undergraduate education.

Keeping pace with the development of time-sharing systems has been the creation of new, "conversational" computer languages, such as BASIC, which are very simple, user-oriented and easily learned. BASIC is the most popular and has made it possible for more students and faculty to learn to program and use the computer than ever before.

The importance of the new role of the computer and computing in the educational process was quickly recognized. In 1970, the National Science Foundation and the University of Iowa sponsored the first Conference on Computers in the Undergraduate Curricula. A conference has been held each year since that time.

The demand for a college graduate to have some knowledge of computers and/or computer programming is growing, particularly in the areas of economics and business. This demand will eventually reach a point where one will not be considered "liberally" educated without this knowledge. One way to meet this need is to integrate the computer into the undergraduate education by means of computer-supplemented courses.

THE COURSE: This paper is an account of a computer-supplemented course in the mathematics of finance. Such a course is currently being given by the author each semester at Eastern College. Included in the paper are several of the programs used in his course plus a discussion of its advantages and disadvantages. It is offered at the freshman level with the only prerequisite being a good background in high school algebra. No previous experience with computing is required and it can be structured around any mathematics of finance text. The text used for the course was Mathematics of Finance by Cissell and Cissell. Beginning in the fall of 1978, the author will be using his own text, Computer-assisted Mathematics of Finance to be published shortly by University Press of America, Washington, D.C.

Using the BASIC computer language the course blended together the topics of the mathematics of finance with elementary programming. This blend was accomplished by introducing the necessary computing techniques and BASIC statements, as they were needed, to solve various problems from the mathematics of finance.

Simple interest problems were used to introduce such BASIC statements as LET, PRINT, and END. During the first class period the concept of inerest was discussed and the simple interest formulas I=Prt and A=P(1+rt) were derived. The first formula gives the simple interest on an investment of P dollars at an interest rate of r for a time period of t years. The quantity A is the future value of P after t years. When the students returned for the next class, after having completed an assignment consisting of various simple interest problems, they were shown how one could program the computer to solve the same problems. For example, to find the simple interest on $535 at 5½% for 3 months, one could use the following program.

```
10 LET P=535
20 LET R=.055
30 T=3/12
40 I=P*R*T
50 PRINT"THE INTEREST IS",I
60 END
```

Each student was required to enter this program into the computer and run it. The purpose of this exercise was to familiarize the student with the computer system. It required each student to learn several system commands and how to sign on and off the computer.

To introduce looping and to demonstrate the interactive capability of the BASIC language, the following program was used.

```
20 PRINT"TYPE IN P, T, AND R(DECIMAL)"
30 INPUT P,T,R
40 I=P*R*T
50 I=INT(I*100+.5)/100
55 PRINT"THE SIMPLE INTEREST IS",I
60 GOTO 30
70 END
```

This program also introduces the built-in BASIC functions. In line 50 the function INT is used to round off the interest to two decimal places.

A careful study of this program reveals that it contains an infinite loop. The students quickly discover this when they run it. This led very nicely to a discussion of the IF/THEN statement. The addition of the line

```
35 IF P=0 THEN 70
```

will provide a means for exiting from the loop by entering zero for P on line 30.

The READ and DATA statements were demonstrated by means of the following program.

```
 10 READ P,R,T
 20 I=P*R*T
 30 I=INT(I*100+.5)/100
 40 PRINT"THE SIMPLE INTEREST IS",I
 50 GOTO 10
 60 DATA 1000,.05,.25
 70 DATA 535,.047,.5
 80 DATA 78.46,.065,2.5
 90 DATA 5000,.0718,.75
100 END
```

Programming the compound amount formula led to a discussion of the operator hierarchy and the operation of exponentiation. The following program was used.

```
10 P=750
20 J=.0538
30 K=12
40 N=5
50 I=J/K
60 S=P*(1+I)**(K*N)
70 PRINT"THE COMPOUND AMOUNT IS";S
80 END
```

In order to conveniently include the formulas for $s_{\overline{n}|i}$, the future value of $1 per period, or the formula for $a_{\overline{n}|i}$, the present value of $1 per period, in a program user-defined functions were used. Subroutines were useful for a program to find the interest rate in an ordinary annuity problem.

FOR/NEXT loops proved to be very useful in a program used to construct an amortization schedule. This program also made use of string variables. The complete program follows.

```
100 F$="###     ###.##     ####.##     ####.##"
110 PRINT"TYPE IN PRINCIPAL AND PAYMENT",
120 INPUT P,R
130 PRINT"TYPE IN RATE AND # OF PAYMENTS"
140 INPUT I,K
150 FOR N=1 TO K
160 T=P*I
170 V=R-T
180 P=P-V
190 PRINT USING F$,N,T,V,P
200 NEXT N
210 END
```

One of the most challenging programs for the course was the program to find the yield rate for a bond. This program uses a sophisticated mathematical method known as <u>interval halving</u>. Consequently, it is too long and complex to be included here, but may be found in the author's book. The reason this problem is difficult is because there is no way to isolate the variable i in the formula for the value of a bond. The formula is

$$V=C(1+i)^{-n}+Fra_{\overline{n}|i}$$

where $a_{\overline{n}|i}=(1-(1+i)^{-n})/i$.

The topics of the mathematics of finance gave rise to many excellant computer exercises. The following is a list of just a few.

1. Write a program to compute net cash flows for various methods of depreciation-cash flow analysis.
2. Write a program which solves for any unknown in an annuity or compound interest problem given the other variables.
3. Write a program to print a bond schedule or table.
4. Write a program to create a table of accumulation factors for some unusual interest rate.
5. Write a program to find the exact time between any two dates.

The general pattern of the course was to develop the mathematics of finance and the necessary elements of BASIC programming side by side. The latter to enable the student to solve each new type of problem using the computer. Eventually the students were given an opportunity to write their own programs. Almost every topic usually covered in a course on programming in

BASIC can at least be introduced through this course.

ADVANTAGES AND DISADVANTAGES: One of the best ways to learn mathematics is to *do* mathematics, i.e. to solve many, many problems. Repeated use of formulas helps a student to memorize them. In the course outlined above, this is accomplished by first solving a problem by hand (or calculator) and then by solving the same or similar problem by computer. Writing a formula, say for the present value of an annuity, in a computer program forces the student to pay close attention to the arithmetic operations involved. He clearly sees the need for the parentheses in certain formulas. Writing computer programs strengthens his understanding of certain algorithms or processes. The precision required in order to solve the problems by computer helps the student to sharpen their knowledge of mathematics.

Solving problems with the computer eliminates the need for interpolation which is a time consuming process and difficult for many students. Besides, the computer offers greater accuracy than interpolation.

Since all mathematics of finance problems involve some rate of interest, their solution requires the use of a mathematical table corresponding to the rate involved. However, interest rates are constantly changing in real life situations making it difficult, if not impossible, to have a table for the rate desired. With the computer no table is necessary allowing any interest rate to be used in a problem.

The course succeeds in giving each student a chance to use a computer; something many never even dream of doing. It clearly furnishes them with a good introduction to computer programming making it easier to succeed in a programming course. It gives each student exposure to a computer which they might never have had otherwise.

Using the computer-both in and out of class-breaks the monotony and removes some of the dullness of the traditional lecture approach. It generates more interaction between the students and the instructor as they try to relate the computer to the theoretical discussions in the text as well as in class. Students gain a much better understanding of the formulas then would be possible otherwise, and it removes their fear of machinery.

In general, the student response to the course was quite favorable. Many were surprised how easily computing with BASIC could be learned.

There were two major disadvantages to using the computer. The first had to do with class time. Using class time to discuss BASIC and computing cut into the amount of time spent on the mathematics of finance. Secondly, it was necessary to duplicate large amounts of material on BASIC to supplement the textbook for the course. However, this problem will be eliminated by adopting the aforementioned book by the author as the text.

The course described in this paper should not take the place of a good programming course. It should remain merely as a substantial introduction to computing.

REFERENCES:

1. Cissell, Robert and Helen, *Mathematics of Finance*, Houghton Mifflin Co., Boston, Mass. (1973)
2. Barron, Jonathan C., *Computer-assisted Mathematics of Finance*, University Press of America, 4710 Auth Place, S.E., Washington, D.C. 20023, (1978)

IMPACT OF CAI FOR REMEDIATING BASIC SKILLS DEFICIENCIES OF OPEN ADMISSION STUDENTS

T. Barrozo, Ph.D.
Director CAI Center

H. Richards, Ed.D.
Associate Dean of Academic Affairs

H. Olsen, Ph.D.
Professor of Social Science

Medgar Evers College, City University of New York

The purpose of this study was to consider as to whether pre and/or currently enrolled open admission college students can be taught basic skills (reading, writing, and mathematics) via Computer Assisted Instruction (CAI) as effectively as through traditional tutorial methods. Analysis of data indicates that the gain in reading scores for 231 CAI students (2.42) was significantly better ($p. \leq .001$, $t = 6.93$) than for 71 non-CAI students (1.90). Data analysis of writing scores illustrates that the CAI students made a significantly larger gain than the non-CAI students in writing proficiency. Although the gain in mathematics, 1.48, was significant for 124 CAI and 182 non-CAI students there was no significant difference in the gain between CAI (1.50) and non-CAI (1.46) students.

Introduction

Use of computers for either Computer Assisted Instruction (CAI) or Computer Managed Instruction (CMI) in undergraduate teaching is in its infant phase of growth. Computers have been utilized to augment classroom instruction, as supplementary "drill-and practice," and autonomous instructional media. Jamison, et. al. (1974), in their survey, cited numerous studies that used CAI as both tutorial and independent instruction. Rossale (1968), employed CAI with second year medical students in Cardiology Endocrinology and Metabolism, Gastroenterology, Bone and Joint Diseases, and Pulmonary Diseases. Results indicated that CAI students scored as well on a final cardiology exam as did a control sample of practicing internists. Hunka, et. al. (1975); McEwen and Robinson (1976); Kearnsley et. al. (1975) have all reported similiar findings in the areas of Secondary School French and for Guidance and Counseling practice. However, it should be noted that little, if any, research has been reported analyzing the impact of CAI for remedial instruction of college students.

Objectives:

The purpose of this study was to consider as to whether pre and/or currently enrolled open admission college students can be taught basic skills (reading, writing, and mathematics) via Computer Assisted Instruction (CAI) as effectively as through traditional tutorial methods. Thus, this study was designed to provide answers to the following questions:

(1) Do the pre and/or currently enrolled open admission students improve their reading ability through "drill and practice" techniques through CAI? Through attendance in the Reading Laboratory? Is one method more effective than the other?

(2) Do the pre and/or currently enrolled open admission students improve their ability in grammar through "drill and practice" techniques through CAI? Through attendance in the Writing Laboratory? Is one method more effective than the other?

(3) Do the pre and/or currently enrolled open admission students improve their mathematics ability through "drill and practice" techniques through CAI? Through attendance in Mathematics Laboratory? Is one method more effective than the other?

Method:

Each student (approximately 950) entering City University of New York takes a battery of College Placement Tests for remediation placement, measuring student progress and academic evaluation. In the Fall of 1976 the battery at Medgar Evers College, one of the nineteen (19) CUNY units, included:

1. Metropolitan Mathematics Achievement Test (mathematics);
2. McGraw-Hill Writing Test (writing);
3. Stanford Reading Test (reading)

On the basis of these pre test results students are exempted or placed in remediation. Students who have a remediation assignment are administered the aforementioned tests again at the end of each semester as a post test.

The CAI students were assigned to the CAI Laboratory to supplement classroom instruction, a "mini-computer" with 16 CRT's. The computer system contains 4 basic curricula: English grammar, Reading, Mahtematics, and GED prepping the curricula themeselves were developed by P. Suppes' staff at the Institute for Mathematical Studies in the Social Sciences at Stanford University. The Writing or Mathematics Laboratory, staffed with professionals and student tutors, to supplement classroom instruction.

Of the 247 students identified as needing reading remediation, 34 were assigned to CAI instruction and 213 were assigned to the Reading Laboratory; of the 229 students reading writing remediation 71 were assigned CAI and 158 were placed in the Writing Laboratory; and, of the 306 students needing mathematics remediation 124 were assigned.

For statistical analysis T-tests were performed on the mean differences between the pretest and post test scores on the various cognitive measures. The T test indicates the probability that the differences between the two mean scores is not a matter of chance.

FINDINGS:

All Medgar Evers College students are administered a battery of standardized examinations upon entering the College. On the basis of these pretest results students are place in or exempted from remediation courses. Students who have a remediation assignment are administered standardized tests once again at the end of the semester; this is the posttest. Movement within the remedial sequence and advancement to college level course work is predicated on these posttest scores. In the Spring of 1977 student "gain" were the principle source of data for an evaluation of the remedial programs and the laboratory components, computer-assisted and traditional.

From the demographic data available some general characteristics of the remediation students emerged. Female students tended to have higher pre-test scores, make more gains, and complete at a higher level than male students in both reading and writing. Male students had higher pre-test scores than the females in mathematics; and while not making as great gains as the females, still ended the semester at a higher post-test score.

Table I

Mean Difference in Grade Equivalent Scores
from Pretest to Posttest
Stanford Reading Test

	HUMANITIES	CORE	ALL STUDENTS
CAI			
Number of Students	33	1	34
Mean Score (difference)	2.48	.30	2.42
Standard Deviation	2.03	.0	2.04
t - Values	t=7.02*	t=0.0	t=6.93*
Non-CAI			
Number of students	132	81	213
Mean Score (difference)	1.79	2.07	1.90
Standard Deviation	1.93	1.56	1.80
t - Values	t=10.68*	t=11.93*	t=15.40
Total			
Number of students	165	82	247
Mean Score (difference)	1.93	2.05	1.97
Standard Deviation	1.96	1.56	1.84
t - Values	t=12.63*	t=11.85*	t=16.84*

* $p \leq .001$

Regardeing Table 1 for the reading remediation students the following remarks appear appropriate:

1. For the 34 students assigned to the CAI, as opposed to the traditional laboratory, there was a mean grade equivalent "growth" between pre and post-test on the Stanford of 2.42 ($p \leq .001$, t-value = 6.93)
2. For the 213 NON-CAI students the mean "growth" was 1.90 ($p \leq .001$, t-value = 15.40)
3. CAI students "grew" .52 grade equivalents more than NON-CAI students;
4. The 165 Humanities Division reading students "grew" 1.93 and SEEK Program reading (82) "grew" 2.05, a difference of .12 grade equivalents, a non-significant difference.
5. Overall CAI and non-CAI students "grew" 1.97 grade equivalents.

The data for the remedial writing students is the following:

Table 2

Mean Standard Score Difference on the McGraw Hill Writing Test

	Humanities	CORE	Total	T-Value
CAI				
Number of Students	69	2	71	
Mean Score (difference)	4.04	1.00	3.96	4.46*
Standard Deviation	7.47	9.90	7.48	
Non-CAI				
Number of Students	91	67	158	
Mean Score (difference)	2.45	.93	1.80	3.69*
Standard Deviation	6.59	5.43	6.15	
Total				
Number of Students	160	69	229	
Mean Score (difference)	3.14	.93	2.47	5.63*
Standard Deviation	7.00	5.48	6.65	

$P \leq .001$

Regarding Table 2 the following remarks are warranted:

1. The CAI students show a mean standard score difference of 3.96 as compared with 1.80 for the NON-CAI students a difference of 2.16;
2. The Humanities remedial writing students did significantly better than the Special Program students in both the CAI and NON-CAI groupings; and
3. Overall the mean standard score difference between pre and post-test is 2.47.

For remedial mathematics students we have:

Table 3

Mean Grade Equivalent Difference Between Pretest and Posttest on the Metropolitan Mathematics Achievement Test (Intermediate Level)

CAI	Computation	Concepts	Problem Solving	Total Score
Number of Students	95	93	80	124
Mean Score (difference)	0.88	1.35	1.67	1.50
Standard deviation	1.32	1.34	1.85	1.30
	t=6.52**	t=9.72**	t=8.09**	t=12.89**
Non-CAI				
Number of Students	123	121	95	182
Mean Score (difference)	1.03	1.28	1.76	1.46
Standard Deviation	1.20	1.26	2.07	1.10
	t=9.57**	t=11.22**	t=8.30*	t=18.01**
Total				
Number of Students	218	214	1.75	306
Mean Score (difference)	0.95	1.31	1.72	1.48
Standard Deviation	1,25	1.29	1.97	1.81
	t=11.41**	t=14.86**	t=11.57**	t=21.52**
t-CAI & Non-CAI	-0.85*	0.37*	-0.32*	0.27*

*$p \geq .05$
**$p \leq .001$

1. While there are 124 CAI students, 1[illegible] NON-CAI students, and total of 306 having pre and posttests, the corresponding scores for the parts of the test are not always available;

2. Since the t-scores are high the mean difference scores for both CAI and NON-CAI are statistically significant $\leq .001$

3. There is a .4 difference between the CAI and NON-CAI students.

To a limited degree it was possible to examine the educational impact of individual, faculty teaching strategies. This sort of analysis could be valuable to the college insofar as it isolates and identifies characteristics of "successful" faculty.

As mentioned earlier, during the Spring of 1977 there were available on the computer two levels, Basic and GED (intermediate), of curricular material. If "successful" CAI student is defined as any student scoring above the mean "growth," then the following represent an analysis of the appropriateness of the available curricular CAI material for dealing with remedial problems in the areas of reading, mathematics, and writing:

1. The basic READ curriculum takes a considerable amount of time to complete but the amount of time spent on the curriculum is not commensurate with the gains;

2. The more time spent by the reading student on the GED reading curriculum, the more successful the student;

3. The "successful" reading students completed more of the GED curriculum lessons with a score of 75% or better;

4. The "successful" writing student spent more time on the GED English Curriculum than his "unsuccessful" counterpart;

5. The "successful" writing student completed with a score of 75% or better more GED English lessons;

6. There appears to be no correlation between amount of time spent on the GED Math Curriculum and "success;"

7. There appears to be no correlation between the number of GED Math lessons completed and "success."

On the basis of these findings curricular prescriptions for the academic year 1977-78 have been decided upon which make maximal use of student time and available resources. Basic reading, writing, and mathematics CAI curricula will be assigned in the future only to CAI GED students; intermediate (GED) curricula in all three areas will be used extensively for the remedial students and pre-college curricula such as those for reading and writing will be experimented with during the academic year 1977-78.

Six and twelve weeks performance and attendance alerts are sent to students where appropriate. The study appears to confirm that these alerts serve an important function, that there is an interesting statistical correlations between student performance and these alerts. Generally, what is true of Humanities Division reading or writing remedial students is true as well of Special Programs and Mathematics remediation students:

A. Attendance: During the first six weeks the "better" students attend class less frequently: but during the first 12 weeks of the semester the larger "gains" are the reward for students who attend class.

B. Performance: For the reading students performance alerts correspond rather nicely with student performance in the course, i.e., performance on the Stanford. The 6th week performance alert for writing students apparently is not discriminating: The "better" students were placed on alert; this situation was corrected by the 12th week.

For the academic year 1977-78 (and as a result of the research done on the Spring of 1977) The following desiderata emerge for purposes of further evaluation and refinement of the use of CAI in the remediation program.

1. Again pre and posttest results for all remediation students must be on file for determining student "growth" and for purposes of advancement through the remedial sequences;

2. Whenever possible an instructor should be assigned multiple sections of a course so that some CAI sections and some NON-CAI section have the same instructor;

3. More students should be assigned to the CAI laboratory in order that a significant number of students are monitored in the CAI laboratory environment;

4. An attempt should be made to see whether there are some remedial problems which can be handled on the machine without any classroom instruction;

5. Efforts shoud be made to integrate the remedial laboratory resources - students might be sent to one laboratory for one purpose and another laboratory for a different purpose.

Generally, and separate from experimentation with CAI, future evaluations of remediation at Medgar Evers should attempt to discover the strengths and weaknesses of programs, whether in Humanities, Special Programs, or Natural Science. It also appears desirable to pursue the question of effective teaching in order that the more "successful" teachers could share some of their knowledge with other faculty.

A COMPUTER-BASED LEARNING AID IN ELECTRICITY AND MAGNETISM

Alfred Bork
Department of Physics
University of California
Irvine, California 92717

An interactive computer-based dialog, developed at the University of California, Irvine, is a complete learning module for electrostatics, at the beginning level.

Introduction

This paper describes an interactive computer program developed by the Physics Computer Development Project at the University of California, Irvine, for assisting with the learning of electrostatics. The program was developed with support from the National Science Foundation, which has supported in various ways the efforts of the Physics Computer Development Project for ten years, and with support from the University of California.

A multicampus grant for four University of California campuses (Irvine, Santa Cruz, Riverside, and Davis), from instructional improvement funds available within the University of California, allowed faculty members on a number of other campuses to work with us in the development of computer dialogs. The particular dialog being discussed, FIELD, involved participation by the following individuals: Robert Eisberg (Santa Barbara), Bruce Rosenblum (Santa Cruz), Sun-Yiu Fung (Riverside), Richard Ballard (Irvine), Joseph Marasco (Irvine), and Alfred Bork (Irvine). It was developed in stages over a considerable period of time, and only gradually reached the form to be described in this paper.

Structure and Capabilities

FIELD is concerned with electrostatics, as it is taught in typical beginning physics courses. It is organized around a central map, a guide to the use of the program. Students can return to the map at any input by simply typing MAP.

Four basic types of facilities are provided in the map. First, a series of learning sequences develops the fundamental ideas of electrostatics. Second, a plotting capability allows students to plot electric field lines and equipotential lines. Third, two built-in quizzes or tests in the program check the student's knowledge of ideas developed within the dialog. Finally, the map allows the student to exit. The map provides a high degree of learner control, including the opportunity to browse through various sections of the program. Access to the quiz sections of the program is controlled by a course management system, in order to protect the security of the quizzes, but all other sections of the dialog are freely available for student use.

Learning Sequences

As indicated, the dialog provides learning facilities for the student who is studying electrostatics. The five divisions are the magnitude of the force between charges, the direction of the force, the force on several charges, the notion of electric fields, and the notion of electric field lines. The early sequences assume a student who has little background with this type of material. Through the map the student can move rapidly through the material if in fact he or she is already familiar with the basic ideas.

As with other computer dialogs developed at Irvine, the student engages in a "conversation" with the computer. Typically the computer presents information, both verbally and pictorially, and is asking questions about the information presented. As with our other dialogs, FIELD stresses graphic learning modes. The student types in responses, and these responses are analyzed by the computer program. Depending on the responses, the program can then go off in a number of directions. The situation is similar to other use of the computer with computer-based learning material, not only at

Irvine but elsewhere. But since not all readers may be fully aware of it, it has been described briefly.

We attempt to deal with as wide a range of student input as possible. Furthermore, some unique capabilities are present that are often not present in teaching programs, based partially on the fact that there is an accurate pointing device available for use on the displays that we use with students, Tektronix 4013s.

Great care is used in building toward the fundamental idea of electric field, as it is felt that this powerful idea, critical in modern science, is often ill understood by students in beginning physics courses. In introducing the idea several small simulations are incorporated within the computer dialog. Courses often talk about the use of unit test charges, and that concept is used as a gateway to the field notion. Here we have the advantage that we can let students "have" a unit charge, simulated on the computer, let them put it at any location, and then can show the force that would be on the unit charge at that particular location.

We start the sequence about electric field with two given charges, present on the screen, and allow the student as many opportunities as desired to move the test charge to new locations; we show the force arrow at each of these locations. Then, to emphasize that it is not critical that we know where the charges are, we cover up the region with charges. Again the student is provided with a unit test charge. Thus, the notion is built up, through actual experience rather than by just verbal discussion, that a force acts on the test charge no matter where it is placed. This notion then leads naturally to the concept of electric field.

The next critical idea developed with the dialog is that of electric field lines. These visual representations of the electric field form, it is generally agreed, an important technique for gaining insight into the way electric fields behave. Thus, field line plots are used in almost every beginning textbook. Again students often, it is found, do not understand these field line diagrams. It is of central importance in the present computer learning material that this notion be explained very clearly, that we provide practice for students in using the notion, and that, as will be seen later, we test to see that the concept is actually understood.

The initial introductory path introducing the idea of field lines is again that which is similar to that in most courses; but once again we provide an active experience for students rather than a passive experience of describing what happened. Thus, we can have the student, with the use of the built-in pointing device, use a test charge to begin to "follow" a field line. We can also have the student relate the field line to where the charges are located and to the sign of the charges.

Field Line Plotting

One of the major roles that the computer can play in beginning science and mathematics courses is that of assisting students in building up intuition, by providing a rich and varied collection of experiences for the student that is not available in the ordinary world. We have developed a number of such controllable worlds, providing the student flexible capabilities for gaining such experience. Here the calculation capability of the computer plays an important role.

The plotting capabilities of FIELD allow the student to place charges of any magnitude at any location on the screen. Then the student can request a field line plot based on the charges just specified. Charges can be added or removed as desired.

The student can also explore a particular interesting region with a particular collection of charges by requesting that a field line be drawn through a designated point, again using the built-in pointing device available. Thus, in a region where the field is zero it is interesting to note the pattern of field lines as they "avoid" such a region.

The program also allows the plotting of equipotential lines, allowing another view as to how the electric field is behaving in the region. These are shown as dotted lines, to distinguish them on the same plot from the field lines. Again, the pointing mechanism is used to specify where the lines are to be drawn, so the drawing is under the control of the student.

Great care is taken in computing the field lines and the equipotential lines. A Hamming predictor corrector method is used to solve the necessary differential equations. It was pointed out to the author some years ago, by Tim Kelley of Southern Oregon College, that the use of simple integration techniques to electric field lines often leads to noticeably incorrect results. Hence, a sophisticated technique such as that employed in FIELD is required. Unfortunately a consequence of this requirement is that when the computer has a heavy load, as it will often have in the middle of the day toward the end of the quarter, the field line plots will appear rather slowly.

Quizzes

Two quizzes are currently available within the FIELD idalog. They do not cover all aspects of the learning material, and it is hoped that at some future time it may be possible to add additional quizzes.

These quizzes are taken directly at the terminal by the student. As they tie in with a full course management system, along with many other quizzes, a brief description of that system will be provided. The system has been described in several papers, available from the author.

The course management system is responsible for controlling access to the tests and for recording the results of the tests. As these tests are typically used in a self-paced, PSI, environment, they must be completed at the 90% level for the student to pass. The student who does not pass a test on a first occasion studies further and takes the test again. A time of twelve hours must elapse between two successive attempts of the same test by the same student. Each test is unique; a variety of techniques are used to generate different questions for each student. Tests have a maximum 40 minute time allowed at the terminal. As with other PSI environments, a student cannot take a given test until he or she has completed earlier required tests. All these facets of access and record keeping are controlled by the computer itself.

The first test tests the basic notions of electrostatics with regard to forces and fields. The student is given two charges, of different magnitude. The problem is to find where the field is zero. As with many of our other tests an optional help sequence is provided for aid with the problem.

The second test is, we believe, a more unique type of testing, not seen in the the usual beginning course. In this test we try to determine whether a student understands the concept of electric field lines. We check to see whether the notion of the direction of the field line is associated in the student's mind with the direction of the field at the same location, and we check to see whether the student has reasonable views as to the density of the field lines and the strength of the electric field. Primarily, however, this quiz is concerned with whether the student can relate electric field line patterns to the charges which produce these patterns. Thus, in one sequence electric field line plots are presented with a hole in the middle, with the charges and the field lines close to the charges missing. Then various charge configurations are presented as possibilities for the problem, and the student is asked in turn whether each charge configuration agrees with the field line pattern. As with other aspects of the quizzes, these patterns are picked randomly from a large collection.

Environment

This dialog, as with the other dialogs available at Irvine, are run on a medium-scale timesharing system, a Honeywell Sigma 7 under the CP/V operating system. About 25 Tektronix 4013 displays are available for student use; both the computer and the displays are shared with other educational research and administrative activities on the campus. Irvine also operates two other central computing systems, a DEC System 10 and a PDP 11/45.

The underlying software for implementing the dialog is a macro-based system developed within the Physics Computer Development Project, particularly for programming interactive computer dialogs. Calculational aspects of the program are done with FORTRAN subroutines. Authoring aids are available within APL to assist with the graphic design and construction aspects of the dialog. COBOL is used for programs associated with the course management system, and the database for the system is a Codasyl-type database. Aspects of the software have been moved to a DEC System 10. Information about these and other aspects of our project can be obtained from the author.

CONFERENCE ON INTELLIGENT VIDEODISC SYSTEMS

Alfred Bork, University of California, Irvine
Arthur Luehrmann, University of California, Berkeley
Edward Schneider, Air Force Human Resource Lab, Denver

A recent conference on combining computers and optical videodiscs was sponsored by the National Science Foundation.

The conference on intelligent videodisc systems was held at Pajaro Dunes on November 30 to December 3, 1977. The following summary, prepared shortly after the meeting, is intended only as an informal guide to the activities of the meeting. More detailed information will be available on request. The conference was supported by the National Science Foundation.

Purpose of Conference

The organizers proposed the conference because they feel that an important moment in education is occurring, the moment when the computer plus powerful associated audiovisual capabilities can have a major effect on education at all levels. The computer has already proved to be a powerful aid to learning, and its steady decrease in price makes it more and more attractive economically. The optical videodisc provides a versatile low-cost storage medium for both audiovisual material and for computer code and data. Further, it is not erasable, hard to copy, and requires no special handling. The combination of these two technologies has exceptional promise for education.

The principal purpose of the conference was to formulate a plan for development not only of the necessary hardware but also the possible production of materials to be available on a commercial basis. The system would involve a combination of the latest computer technology and the latest videodisc technology.

The people invited to the conference were carefully chosen. At least four types of individuals were represented. First, there were people from academic institutions, and similar organizations, who are particularly concerned with using the computer as an educational tool. Second, there were the developers and potential manufacturers of the videodisc technology. Third, there were the developers and producers of computer equipment. Finally, representatives from a number of governmental agencies were present. Forty-four people attended.

A series of demonstrations was available for participants to view at times during the conference. These demonstrations will be described in more detail later in this report.

First Day

The conference began with three informal review papers. As many of the people in the audience were experts in the three areas involved, they were invited, and sometimes called on explicitly to add to these introductions.

The first discussion was by Arthur Luehrmann, "Trends in Private Computing." He surveyed current personal computers and projected cost estimates in a variety of ways of stand-alone computer systems. The second talk, "State of Educational Computing and Its Requirements," was given by Alfred Bork. He reviewed the ways the computer could be used in education, a few projects, hardware requirements (including the need for graphics), and the problems of production in publishing such materials. The third talk, "Videodiscs as Home Entertainment Media and Other Prospects," was given by Edward Schneider. It reviewed the videodisc technology as it has been developed by a number of different vendors, and also reviewed the cost of putting visual and film material on such a medium as compared to the cost of putting the same material on other media.

In the afternoon session on the first day the participants discussed three further items, all considered in more detail at later stages of the meeting. The first topic was that of the time scale. How soon is it likely that computer-based learning material, including computer-based learning material involving the videodisc technology also, will become widespread in American universities? Some feeling was expressed that a Carnegie Commission report, The Fourth Technology - Instructional Revolution in Higher Education, offered a time scale which was reasonable, although a bit conservative. Here it was predicted that between 1990 and 2000 such material will be introduced very rapidly, and that after 2000 it will be generally available. The Carnegie report refers to higher education, but no argument was offered that a different time scale would apply in any other area.

The hardware situation was indicated by some of the demonstrations available at the conference. Two videodiscs were available, the Teldek disc and the Thomson-CSF Optical Video Player. Only optical players seemed to fit in with the educational requirements. There has been a long series of delays in the introduction of the home optical videodisc system, with test marketing of the Philips system now announced for this spring. But a commercial-industrial educational system is available now from Thomson-CSF and MCA. These two systems have been widely shown, particularly in the military environment, and they are currently operational. Limited production of the Thomson-CSF system will begin soon in France. Initial units are, in the small scale production, estimated as costing approximately $4,000.

A variety of state-of-the-art computer systems were also available. A full-scale, high resolution, graphic display from Tektronix, was used to run some demonstrations of computer-based learning material. A stand-alone system, with only slightly less resolution, with floppy disc and powerful calculating capabilities, costing about $5,500, was exhibited also; this system is sold by Terak. Finally, two of the initial entries into the home computer market (as opposed to the hobbyist computer market) were available, the Commodore PET machine and the Radio Shack computer; both of these systems sell for approximately $600. While the Commodore PET and Radio Shack machines were considered by most of the participants at the conference as inadequate for current educational uses, they indicate the rapid decrease in the cost of such hardware. Other vendors are expected to enter this market, with increased capability.

An important issue in the afternoon and evening sessions was that of courseware production. It is realized that the hardware alone is not sufficient, but that hardware sales will be driven by the availability of well tested, effective course materials. A vigorous discussion ensued as to the ingredients of a production method, with participants describing successful systems for producing computer-based learning materials and systems for producing videodisc materials. The question of markets also received attention in these sessions.

Second Day

The second day was devoted to setting the stage for several smaller working groups, followed by the meetings of the groups. These groups were asked to consider the evolutionary paths that might lead from our present situation to the widespread use of intelligent videodisc systems, considering all the problems involved. They were invited to develop scenarios as to the stages along such development. A four to five year time scale was suggested. As of this writing, the codirectors have not had an opportunity to fully integrate the reports of the groups. The following outline shows the range of the activities considered.

The groups discussed seven stages in the evolution toward commercial production of learning material involving computers and videodiscs,

1. Prototype courseware development projects
2. Development of hardware for delivery system
3. Hardware development for authoring system
4. Study of market possibilities
5. Development of production systems
6. Courseware production
7. New organization mechanisms

Third Day

The short third day began with a discussion of the reports of the working groups, and ended with a summary prepared by Joseph Lipson.

THE WHENS AND HOWS OF
COMPUTER BASED
INSTRUCTIONAL SIMULATION

Bobby R. Brown
and
Robert S. Ellinger
University of Iowa

Computer Assisted Instruction (CAI) and in particular that subgroup of CAI called Computer Based Instructional Simulation (CBIS) can be an effective tool for teaching certain types of concepts. This is particularly true for instructional simulation. This paper will address three questions. First, what makes a good CBIS? Second, for what types of learning is CBIS best suited, when is the CBIS doing its job? Third, how do you evaluate the job it's doing?

INTRODUCTION

Computer Assisted Instruction (CAI) and in particular that subgroup of CAI called Computer Based Instructional Simulation (CBIS) can be an effective tool for teaching certain types of concepts. To date, however, most studies comparing CBIS with other teaching methods have measured aspects of the student's performance for which the traditional teaching forms would be expected to perform at their best e.g. factual recall and concept definition. This is due in large measure to the fact that current evaluation methods are best suited to evaluating the results of these types of learning. It has long been recognized that CBIS does have unique capabilities for teaching; often these capabilities have been referred to as desirable but unassessed benefits (Edwards).

This paper will specify these benefits and test for them in a CBIS package. In order to assess the potential benefits of CBIS, we will address three questions. First, what makes a good CBIS? Second, for what types of learning is CBIS best suited? Third, how should you evaluate the effectiveness of CBIS? We will suggest testing methods and demonstrate their use on two CBIS packages.

GENERAL PROPERTIES OF CAI

Many properties of CAI have been touted as desirable features of an instructional system. Among these are student motivation, feedback, error detection and correction, reproducibility of teaching, the lack of perturbability of the computer, freedom of students to work at their own pace, the individualization of instruction, the ability to improve the way the material is taught, and the record-keeping ability of the computer (Inbar and Stoll). While we do feel that all of these attributes are important and are achievable on a fully developed CAI package, CAI packages to date have not made proper use of even these features because of the inexperience of program authors and lack of guidelines for them; this has led to much justified criticism of CAI.

One potential benefit of CAI was not mentioned in the list above; the potential for providing enhanced "intuitive grasp" of concepts and interrelationships. Alfred Bork (1977) notes that:

> A complaint often registered at doctoral examinations is that some graduate students can carry out the mathematical manipulations, even for very advanced problems, but have little physical intuition as to what is happening.

He goes on to suggest that CBIS or "controlled worlds" will give the student the experience and insight necessary to gain the intuition to understand the problem.

When: Effective Instructional Design for CBIS

As has been suggested, CBIS is exceptionally effective for teaching certain types of concepts and quite unsuitable for teaching other types. CBIS is particularly well suited for teaching process concepts, the interrelationships between process concepts, and multifaceted/multi-step problem solving.

Process Concepts

Process concepts are conceptualizations of processes. Normally, these processes take place through time and across space. Process concepts are at the highest level of cognitive learning and the student needs the maximum amount of assistance. This type of concept is difficult to teach using traditional methods because the processes cannot be shown in action - when they can, as in a movie, the student has no control over the independent variables nor the environment. In physics, chemistry, and biology the laboratory has been used to supplement the material taught in the classroom, so that the student gains a greater understanding of the process by experience - the experiment. Pedagogic experiments are performed with the student "controlling" the variables so that he can see the cause-effect relationship and the processes in action. Unfortunately many processes take too long, are too subtle to be noticed, or are too cosmic or too microscopic in nature to be brought into the laboratory. But, as Bork noted, the intuitive understanding of these concepts can be extremely important. Further, the social sciences face another problem; the objects of interest are the economy, the political and social systems, and the people of a region. And students cannot experiment with these processes - only politicians can. For these types of concepts CBIS can fill the laboratory-learning gap.

CBIS has many of the positive instructional attributes of well planned lab work, plus certain advantages over laboratory experiments. Depending on the type of terminal - teletype or teletype emulator CRT, semi-graphics CRT, graphics CRT, or color graphics CRT - a CBIS can come extremely close to duplicating the feel of the laboratory situation for the student. The advantages of CBIS include: the enhancement of certain features of the experimental experience so that these features are perceived the first time through; the ability to speed up or slow down time; and the ability to enlarge or contract the space that the experiment requires, so that the patterns and processes can be easily perceived. Thus, CBIS can be used to great advantage for teaching process concepts because it can focus on and enhance the image of a process as it is simulated.

Interrelationships Between Concepts

The relationships between concepts are also difficult to teach, especially when the concepts involve processes. Relationships between concepts are difficult to teach using traditional methods because traditional teaching methods are static, so cause-effect relationships are difficult to demonstrate. While the demonstration of the general form of relationships is possible using audio-visual aids, the student will still have difficulty coming to an understanding of the magnitude of the relationship. This is the second type of concept for which CBIS is ideally suited. CBIS allows the student to guide or use the independent variables to predict variables. When the student creates the causes and sees the effects, the magnitude of the relationship should become apparent. And by understanding the magnitude of the relationship he should be able to predict the effect that given values of an independent variable will have on the dependent variable and to choose values of independent variables that will produce a given effect on the dependent variables - two ways that a student can demonstrate a clear and accurate understanding of the relationship between the variables.

Multifaceted/Multi-step Problem Solving

Multifaceted/multi-step problem solving is the third type of learning that is properly taught using CBIS. Problem solving techniques, particularly when the problems have multiple variables, or use multiple step procedures, are among the most difficult types of procedures to teach because they involve both cognitive and behavioral learning processes. CBIS can provide the controlled environment in which the student can learn part or all of the problem solving techniques that he will need to know before he actually is faced with the problem.

Multifaceted/multi-step problems are those that usually require repeated practice to learn to solve, or those techniques that students "learn by doing"; examples of such problems include learning to play in competitive sports, drive a car, fly an aircraft, handle complex machinery, diagnose and repair communications equipment, or design and construct lighting for plays. For several of these examples CBIS has long been used. Learning emergency procedures on aircraft is a prime example - it would be bad for business to lose several Boeing 747's while pilots were practicing in them. Other examples, while less dramatic, can be cited to illustrate the use of CBIS to help the "students" learn problem solving techniques. Diagnostic techniques are being taught to medical students in preparation for their practice on patients. Law students can practice preparing and presenting briefs and cases. Business students in accounting, management and marketing use CBIS to practice their decision making. In all cases multiple variables need to be accounted for, and there are multiple problems to be solved.

Since the early 1960's, many business schools have used non-interactive simulations, like the "Executive Game" to teach their MBA students management techniques in a competitive business situation. The students compete against one another with profit being the main criterion for deciding success or failure. Since many of these games have since been written or rewritten in interactive simulations, e. g. Market, they must be valuable to teaching management procedures. These management procedures that are being taught are of the multifaceted/multi-step variety. In the management games, some variables are controlled by the student and others are not. Independent variables include the price of the goods, the fixed and variable costs associated with producing the goods, research and developments costs, and marketing costs, plus a sundry of other controllable costs depending on the simulation. Independent variables that are not controlled by the student include the prices and advertizing investments of other firms (students) and "random" events like new labor costs and fires in warehouses. This forces the student to make decisions in a realistic manner or face business failure and a low grade in the course. Likewise, it gives him a chance to see how good his problem solving ability is when compared to his peers, so that he can decide whether he should attempt a career in management, and also where his strengths and weaknesses lie.

Simulations designed to help the student learn diagnostic, problem-solving, or decision-making techniques must have output that is sufficiently realistic so that the student feels he is actually making the decisions or solving the problem while he is running the simulation - otherwise the simulation is at least a partial failure. So for simulation designed for this purpose in particular, the mode of output is extremely important.

HOW: PROPERTIES OF CBIS: THEIR USE AND EVALUATION

The Use of Special Properties of CBIS

When you decide to write a CBIS for teaching, there are several properties of CBIS that you should be aware of. First, CBIS should be used only when it is the best instructional method for the material. Second, good CBIS should use the maximum capabilities of the terminals and computer system. Third, as CBIS becomes complex it should have more documentation. Fourth, a good CBIS package should contain sufficient ancillary material and programming to make learning easy for the student.

Use CBIS Only When Needed

The best known, and by far the most realistic and sophisticated CBIS's, the simulations of every phase of the Apollo moon missions, were constructed for teaching process concepts, the relationships between concepts, and multifaceted problem solving abilities. Astronauts and ground personnel were given every conceivable type of problem to solve that might be encountered during a mission. They repeatedly practiced solving these problems, learning the relationships of the system hardware and software in the process, to the point that when a potential catastrophe occurred in Apollo 13 the CBIS system actually produced the solution to the problem. So first, good CBIS should concentrate on teaching processes or process concepts, their relationships, and multifaceted-problem solving. Other less time-consuming teaching methods should be used when less complex concepts or techniques are taught.

Maximize Capabilities

Second, a good CBIS should make the best possible use of the computer system's and terminal's capabilities. There are two philosophies for attacking the presentation of a simulation: by using standard techniques i.e. making the material "teletype-33" compatible; or by fully using the inherent capabilities of the computer complex on which you are working. The first philosophy is justified by the fact that "ALL terminals" can use the program, making the simulation transportable, or easily transferable from terminal to terminal. This is of great importance on campuses where there are 5 or 6 different types of terminals, and not one of sufficient number to serve a large class. The generally unrealized expense of this type of programming is complete underdevelopment of the visual illustrations and graphic demonstrations so important to good teaching. The importance of these visual illustrations is well documented (Lackmann) and well known to all teachers. For a long time the blackboard (and recently the overhead projector) was the most important teaching aid in the lecture classroom. So, too, the graphic capabilities of the computer complex should be of the highest value to CBIS when used appropriately. Unfortunately, the program complexity increases with the complexity of the visual effects, but this is no excuse for not utilizing the capability of the computer system, any more than lack of ability to draw is a good excuse for not using the blackboard. Nor is the excuse of incompatibility of terminals sufficient because there are simple ways to program so that terminals with similar capabilities but different instruction sets, can use the CBIS materials.

Documentation

Third, while documentation is important in all CBI, it is doubly so in CBIS due to its complexity. It is imperative to document the concepts and techniques and their underlying assumptions in detail. This documentation can be either in the form of computer tutorial material or manuals. If there is one problem that all CAI packages suffer from, it is the lack of adequate documentation. This is particularly true as the CAI packages increase in complexity, like CBIS. Generally what documentation there is, is in the form of a few notes from the author to himself that are undecipherable to any but those as thoroughly familiar with the package as the author. Or a "user's" manual that describes only how to start the program and a little about the concepts taught in a haphazard manner.

Actually there are three types of manuals and two modes for presentation of this material. First, there is the instructor's manual. It should include the objectives the package is designed to teach, the format and structuring of the presentation of the concepts, background information on the concepts, and suggestions for integrating the material into the course. The second manual is the user or student manual that should include any necessary information that the student needs to know before running the package, instructions on how to get the package into operation, instructions on how to get out of trouble, and supplementary information on the origin of the concepts of the package. The third manual is the technical manual. It should explain the purpose and structure of the various programs in the package, lists or legends of the variables, arrays, and pointers or indicators, so that a programmer can make the site-specific modifications required when a complex package is moved to a new computer complex. A detailed description of the organization of these manuals is beyond the scope of this paper but an excellent document of this organization is already available (CONDUIT).

There are two modes available to the author for presentation of the material contained in the manuals, particularly the user manual, printed or interactive. Normally, manuals on how to use something, be it a chainsaw or computer program, come in the form of a booklet, as this is the only economic way to transmit the information. However, the author of the computer program has another tool readily at his disposal, the computer. With it he can present appropriate material and instructions to the user in interactive form. But this brings up the question: what is appropriate material? We believe that appropriate material includes instructions or help units that let the user know what is coming next in the simulation; that explains the details of alternatives when the user is asked to make a choice; or that explains the details and bases of concepts used by the simulation, in tutorial format. Inappropriate material includes the rules of the simulation; supplementary references; long, detailed explanations of the concepts or of the workings of the simulation. While the distinction between appropriate and inappropriate material may be a bit fuzzy, the best rule of thumb would be, if there are more than 24 lines 80 characters long (one normal CRT screen full) of explanatory material without a user response, then the material belongs in a manual. Use interaction, don't write a textbook. And if there is ever a decision between a little more documentation or not, always write more. (It can always be ignored but may by important to someone at sometime for reasons that are not apparent to you.)

Ancillary Material

The fourth property of good CBIS is that it must have sufficient ancillary material, in the forms of both tutorials and manuals, to allow for effective and efficient use of the simulation. The one distinguishing characteristic of instructional simulations is that they are designed to teach complex processes and concepts. They can't do their job properly unless the user understands the simulation and its bases. There are several simulations that would be excellent instructional simulations, but for the fact that the authors failed to realize that the complete simulation would overwhelm the beginning user with too many multiple options.

Good CBIS material provides for these neophytes in one of two ways. First, the simulation can be constructed in several versions, simple to complex, so that the beginner can get a feel for using the simulation before graduating to the next more complex version. In actuality one simulation can be devised that accomplishes this task of creating multiple versions. It is done by asking the user what level he wishes to work at, then defaulting to standard options for the beginner while allowing the advanced user more options. If the user's decision time is an important variable in the simulation, this can be decreased as the level of expertise increases.

The second type of ancillary material CBIS can provide help to the neophyte is to construct a set of tutorial lessons to teach concepts and options to the student. These lessons can include traditional CAI tutorials (programmed instruction, games, and mini-simulations) and sections of the actual simulation, grading to the complete use of the simulation. In practice, both of these teaching techniques can be used together.

Evaluation of CBIS

While CBIS can be an effective tool for teaching certain types of concepts and techniques - given a good understanding of its properties for teaching - the tool has been very difficult to evaluate. Often this difficulty is partly due to lack of understanding of what the simulation is supposed to be teaching and in part to lack of good methodology for evaluating student's performance.

Suppose that a teacher has some material that he thinks can be taught best using CAI. He should only use CBIS if he can demonstrate to himself, at least, that the material falls into one of the three categories described above; process concepts, relationships between concepts, or multi-faceted/multi-step problem solving. Otherwise he should opt for some other type of CAI. One major problem with CAI is that authors of material fit the material to the mode of instruction instead of the mode of instruction to the material, as should be the case. To insure that the author is not using an overkill method of instruction he should write a series of goals and objectives, then decide which would be taught using CBIS and which using other methods. This also will help to cut down on the immense amount of development time required. In writing down his goals and objectives, the author has begun the process of writing the evaluation, that is, he has something that he can test for. But, if the author's material is suitable for CBIS, the author faces the problem that there are few methods available for the evaluation.

Evaluation of CBIS Package: The Salesman on the Beach

This problem was encountered in tests of two CBIS packages written to teach spatial marketing principles to beginning undergraduate human geography students. The first, called "The Salesman on the Beach" (Rushton et al) was developed in the early 1970's to teach the principles of spatial competition in one spatial dimension. The student is told that there are two ice cream salesmen attempting to sell ice cream on a beach and both are attempting to maximize their profits. They can move their ice cream wagons anywhere along the beach - their location is initially the only variable that they can change. Two students represent the two ice cream salesmen who must decide where they should locate to make the most profit. Each student is faced with environmental considerations beyond his control - the location of the population that is non-uniform across space and the other student's location. In later units the students also control the price, which adds another controllable and uncontrolled variable to the problem for each student. This material fits the range of CBIS because the students are learning process concepts, and are learning to solve multifacets problems.

This package was among the first interactive CBIS's in geography and thus the first version was a bit rough because there were no clear principles or properties of CBIS. A second version was constructed attempting to more closely follow the properties described above. However, the hardware limited the attempt - only 300 baud hardcopy terminals were available at the time, so that while the terminals' and system's capabilities were maximized, these were little more than the standard ASCII teletype-33 capabilities. And while the documentation was extensive for the time, with a 13 page student manual and a 27 page instructor-programmer's manual (Rushton et al) the manuals were somewhat deficient. This was due primarily to a lack of guidelines. Also, the program has no ancillary materials though the lesson units were constructed on a scale of graduated complexity. So, it is obvious, at the outset, that this package whould be less than completely successful.

An experiment was run in 1973 to test the utility of this package. It was hypothesized that a group of students using the CBIS package would show a problem solving ability superior to that of a group of students using traditional methods, but that their ability to define concepts would be slightly poorer. The results of the experiment (Table 1) showed these tendencies, but not to a statistically significant degree (Ellinger & Frankland).

Table 1

Student's Test of Difference Between Control and CAI Groups*

	Total	Definition Section	Problem Solving Section
Pretest	1.45	0.21	0.40
Posttest	0.46	0.33	0.65

*after Ellinger and Frankland

The conclusions reached after this experiment were indefinite. First, either instructional method worked equally well. Second, the costs were definitely all in favor of the traditional methodology, but it was recognized that the costs of all CAI would continue to decline so that the cost element would not be a factor (or would favor the CBIS material). Third, many of the students using the CBIS material felt that they learned more and that the material was presented in a more enjoyable fashion than by learning the same material in the traditional manner, but that the CBIS program still had some problems - the chief being the length of time the student had to wait while the program was printing out (this was more a function of the terminal speed than of any inherent programming problem). So, not surprisingly, the experiment was inconclusive.

Evaluation of the Spatial Marketing Simulation

A second experiment has just been completed on another geographic CBIS package. This package, called the "Spatial Marketing Simulation", was developed in 1976-77 to teach descriptive spatial location theory to beginning economic and human geography students. The theory deals with the reasons that firms will choose given locations within a region (the size of a state) to locate, or why firms that chose certain locations within the region were successful or unsuccessful. It is apparent that this theory is much more complex than the model previously described requiring a student to deal with many more economic process concepts and the relationship between these concepts when they interact with the dependent variable. The theory is designed to teach the ability to solve the problems of "where the firm should locate to make the greatest profit or the least loss" and "why some firms are successful and some unsuccessful". This theory is an obvious candidate for a CBIS package, following the ideas suggested above.

The package fully develops the properties described above, making full use of a semi-graphics terminal with cursor addressing, page, line, and element erasure, and many other terminal functions, as well as many software functions like disabling and sensing break. The package is well documented with a 17 page student manual, a 55 page instructor-programmer's guide, and a 15 page user's guide, plus a large number of interactive help units designed to give help at any query in the simulation part of the package. There are also seven lessons, taking approximately 12 hours to complete, designed to teach the students the concepts, relationships, and necessary problem-solving skills to properly use the simulation. This package has all the necessary properties described above to meet the requirements of a good CBIS package.

In the experiment one section of beginning human geography students received instruction via the CBIS package while two sections received instruction via the traditional lecture/discussion/paper and pencil exercises (Ellinger and Brown). A pretest revealed no significant differences between the groups. The results of the posttest are shown in Table 2.

TABLE 2

Evaluation of the Spatial Marketing Simulation

	Total Test		Definitions	
	CBIS*	Control+	CBIS	Control
Mean	11.76	8.19	2.96	2.84
S.D.	2.98	2.66	1.05	.97
t-value	3.68		.406	
Prob.	> .0001		n.s.d.	

	Relationships		Problem Solving	
	CBIS	Control	CBIS	Control
Mean	5.18	3.89	3.75	2.07
S.D.	1.44	1.44	1.58	1.57
t-value	2.84		3.41	
Prob.	> .005		> .001	

* n = 14 + = 37

The mean, S.D., t, and probability values are given for the total posttest scores and for three subscores of interest.

Although significant differences favoring the CBIS group were found on the posttest total score, the subscore results are more interesting. For the "definition" subscore results the group differences are very small, not significant, with the CBIS group mean slightly higher. The absence of significant differences in performance for simple factual recall was expected because this is a known capability of CBIS. Results similar to these are frequently observed in comparisons of instructional simulations with "traditional" instruction.

Without the two additional subscores which, along with the definition subscore constitute the total score, we would not have observed any superior performance attributable to the CBIS group. On the subscores reflecting CBIS's capability to teach, the ability to solve multifaceted/multi-step problems, and the ability to properly relate two or more process concepts, the differences observed between the groups were in the predicted direction.

The contrast in the student performance outcomes of the two experiments above is so marked as to require no further comment. However, some comments on other differences may be worth noting. First, clearly stated instructional objectives were presented for the design, implementation, and evaluation of the "Spatial Marketing Simulation" while derived objectives were stated at the time of revision for "The Salesman on the Beach". Second, the careful development of an evaluation instrument (also based on the instructional objectives) that allowed observation of three types of performance permitted the observation of group differences. The relative lack of precision of the posttest for "The Salesman on the Beach" simulation did not permit the observation of differences in performance. At this time it is impossible to say whether any unassessed differences between the groups existed.

Conclusion

While they are not new or original, we feel that the "whens and hows" discussed here have served us well; we recommend them to you. First, make sure that the material and instructional design require CBIS; second, use the full capabilities of the terminal and computer system; third, document; fourth, use ancillary material when needed to the fullest extent. Then, in your formative evaluation, test for those process concepts, relationships, and problem-solving abilities that you have developed the simulation to teach. For the accomplishment of appropriate instructional tasks, a well designed CBIS package is a powerful tool. Such a CBIS package when properly evaluated should show clear cut observable instructional benefits.

Notes

(1) Teletype-33 programming also increases any deficiencies in the desired features of CAI mentioned above.

Footnotes

Bork, Alfred, "Computers as an Aid to Increasing Physical Intuition", Physics Computer Development Project, (1977), p.3.

CONDUIT, Documentation Guidelines, Trinka Dunnigan ed. CONDUIT C/O Computer Center, University of Iowa (1976).

Edwards, Judith et. al., "How Effective is CAI? A Review of Research", Educational Leadership (1975), 147-153.

Ellinger, Robert S. and Bobby R. Brown, "So You Want to Simulate: Guidelines and Evaluation of Computer Based Instructional Simulation", Submitted to Journal of Geography in Higher Education.

Ellinger, Robert S. and Phillip Frankland, "Computer-Assisted and Lecture Instruction: A Comparative Experiment", Journal of Geography, (1976), pp. 109-120.

Lackmann, "Treading the Troubled Waters of Training: Some Simple Guidelines for Improving Learning Efficiency", Journal of Educational Technology Systems, Vol. 5 (1976), pp. 284-287.

Rushton, G. et. al., "Salesman and the Beach: a CAI Package", Department of Geography, University of Iowa (1973), 27 pages. (Also) Ellinger, Robert S. et. al., Student Manual for Computer Based Instruction Units in Central Place Theory, Department of Geography, University of Iowa, (1972) 13 pages.

COMPUTER-MANAGED
TUTORIALS AT NOTRE DAME

R. E. BURNS
W. I. DAVISSON
C. M. FLANAGAN

UNIVERSITY OF NOTRE DAME

Over the past four years, a group of economists, historians, and instructional materials development specialists at the University of Notre Dame has prepared and placed in operation a series of computer-managed-instruction modules for use in introductory-level courses in Economics and Western Civilization. As developed by the Notre Dame group, a typical tutorial lession consists of about twelve questions presented in a multiple-choice format with programmed responses for each answer delivered over a high-speed data printing terminal. Computer-managed tutorials have become an integral part of the curriculum at Notre Dame; they also have been exported to universities, colleges, and junior colleges throughout the country. More than twenty post-secondary-education institutions with diverse student populations are now using our material.

Introduction

During the last four years, a development group of economists, historians, and instructional materials specialists at the University of Notre Dame has prepared and placed in operation an extensive curriculum of computer-managed-instruction modules for use in introductory-level courses in Economics and Western Civilization.

As developed by the Notre Dame group, a typical CMI module consists of about twelve questions presented in a four-option multiple-choice or true-false format with programmed responses for each answer choice. Each question is preceded by an explanatory lead which helps to establish a context for the question. Leads vary in length from a sentence to a full paragraph. Programmed responses to answer choices may be as short as a sentence or as long as two or three paragraphs. The combination of lead, question, answer choices, and programmed responses is known as a tutorial. Presently, we have a total inventory of about 200 tutorials on line for our Principles of Economics courses and over 600 tutorials operational for our Western Civilization courses. An enormous amount of information is contained within these two inventories of 800 tutorials. Overall, the amount of material written for these two inventories exceeds 2800 double-spaced typewritten pages.

Types of Tutorials

Our inventories of tutorials for the Principles of Economics and Western Civilization courses include three general types of questions: review, new information delivery, and concept analysis. Tutorial questions classified as review relate to subject matter and/or statistical data of sufficient importance to be included in any introductory Principles of Economics or Western Civilization course. In the early years of tutorial writing, our criterion for sufficient importance was consensus - inclusion of material in at least five of the most frequently used college textbooks. However, when the size of our inventories increased and when tutorial selection options were introduced into the programming - thereby enabling faculty users to select from our inventories only those tutorials germane to their personal instructional objectives - we no longer felt obliged to be rigorous about the consensus criterion. We now perceive the subject matter of the tutorials in our inventories to be textbook independent and compatible with most textbooks currently used in introductory Principles of Economics and Western Civilization courses.

Information delivery tutorials are special emphasis items. They relate to subject matter that one would not expect to find in the five leading college textbooks currently used in introductory Economics and Western Civilization courses. Information delivery tutorials are vehicles for meeting the curricular requirements and specific content needs of individual professors. Concept analysis tutorials are concerned with such economic concepts as scarcity, incentives, and gross national product and such historical ones as feudalism, mercantilism, and romanticism. Questions in these tutorials usually are presented in a sequenced series. Students are given definitions of concepts, questioned on their understanding of the definitions, and then presented additional questions with answer choices which require identification of the traits and characteristics of the given definitions in different contextual situations.

Student Access and Programming

Presently, students have access to our CMI tutorials at times convenient to themselves over electronic data printing and CRT display terminals located in supervised labs. Experience has shown that students much prefer terminals with a print capability to those which lack it. Print-outs of tutorial experiences have proven to be extremely valuable study aids, and students are very anxious to obtain them. To date, few courses have been assigned specific terminal lab hours. We have tried to maintain an open-access policy in the labs, but the increasing introduction of CMI segments into courses hitherto taught without them will make the assignment of terminal lab hours to all CMI courses inevitable.

Similarly, the expansion of CMI instructional application and accumulated experience with CMI have forced an evolution of programming strategies. In the beginning, rather than write individual programs for each CMI lesson, the Notre Dame group developed load and driver programs in PL-1 which would enable us to get our materials onto our system and running with minimal difficulty. We soon discovered that other programs were needed which would enable us to modify existing tutorials and to create new ones easily and quickly; they would also allow faculty to select out of our inventory only those tutorials appropriate to their respective instructional objectives. We also discovered that PL-1 limited the exportability of our materials to other institutions. In order to resolve these problems, the Notre Dame group developed a package of new programs known as LESSONWRITER. This package contains three new programs -- CMI, CHNG, and NSELCT -- which are available in PL-1, BASIC, and FORTRAN. CMI accepts input from an interactive terminal to create new tutorials. CHNG corrects spelling and adds or deletes material from existing tutorials. NSELCT enables faculty to select any number of tutorials from our inventory and to organize them into any number of lesson configurations. Faculty can also introduce new tutorials of their own and sequence them appropriately into either existing or new lesson configurations. Once selection of tutorials has been made or new tutorials introduced, NSELCT will number tutorial questions sequentially within a given lesson, while also sequentially numbering the old and new lessons themselves. The LESSONWRITER package as a whole enables us to export our entire inventory of tutorials or selected versions thereof to any educational institution with BASIC, FORTRAN, or PL-1 capabilities.

Faculty and Student Use

At Notre Dame, faculty either have assigned CMI tutorials as continuing post-lecture assignments throughout an entire course or have made tutorials available to students on a purely optional basis. Tutorials have served both as alternatives and complements to discussion sessions, traditional lectures and slide and/or audio-tape presentations; tutorials also have functioned as the principal instructional strategy in a course. In addition, faculty have found the tutorials to be extremely effective basic information sources for the preparation of both objective and subjective course examination questions.

Students are absolutely convinced that time invested in CMI tutorials will improve their grades. Many students appear to be using the tutorials as sources of information rather than as simple review routines. Students go into the terminal labs and deliberately choose incorrect answer choices to get the explanation and information contained in the programmed responses to them. Moreover, feedback from classes suggests that students who in the past found internalization of abstract concepts difficult have discovered that the procedures employed in our concept analysis tutorials and the print-outs of programmed responses function as helpful departure points for inquiring into the nature of such concepts. In effect, the information supplied by the print-outs and the form in which it is presented help to make a given abstract concept more discussable and understandable.

Technical system problems have had some effect upon student satisfaction with our tutorial programs. Some students have complained about slow response time at peak hours of system use, but they remain convinced of their need for what the system can deliver. For the most part, their complaints have been expressions of impatience with the rate of delivery and not with the quality of the material delivered or with the value of CMI tutorials as an instructional strategy.

Evaluating Computer-Managed Instruction

Most attempts to evaluate computer-managed tutorials at Notre Dame have centered on measuring student satisfaction. More than 85% of the students surveyed report high satisfaction with the tutorials and a willingness to recommend other courses using CMI to their friends. An even greater number of students -- 93% -- believe that the time invested in running tutorials results in better scores on exams. More than 80% indicate a willingness to enroll in other courses using computer-managed tutorials, while 78% in fact believe that they learn more from running CMI tutorials than from participating in discussion groups led by graduate assistants. Only 9% of the students surveyed describe the tutorials as impersonal and/or dehumanizing. Some students have complained that the existence and availability of tutorials have made competition for high grades more intense. The tutorials provide such a competitive advantage in exam preparation that grade-conscious students feel obliged to get every bit of information from every answer choice. Consequently, some students spend as much as four or five hours a week on a data printing terminal.

The Future of CMI Tutorials

Although much more evaluation is needed, preliminary data collected at Notre Dame suggest that student performance in any discipline can be improved by CMI potential of CMI tutorials is to be realized and if this mode of instruction is to become widespread among American universities and colleges, tutorials in a variety of disciplines will have to be made readily available at reasonable cost. Moreover, users should be able to select specific contents out of existing inventories of tutorials -- something not possible with print technology, but entirely possible with computer technology -- and expect delivery of such selected materials in forms compatible with the user's computer system. Presently, this kind of selectivity and exportability generally is possible with the limited range of materials produced at Notre Dame. More than twenty post-secondary institutions have incorporated one or more of our CMI programs into their on-going instructional activities. However, at this moment, the inventories of avilable exportable tutorials in the broad range of academic disciplines which expansion of this mode of instruction requires is not available.

In time, inventories of tutorials in most academic disciplines will come. Along with these inventories also will come refinement of the tutorial mode and development of other modes. Whether such inventories will be prepared by university centers, by the data processing industry, or by textbook publishers is problematic. That they will come is much less so; the potential of the technology -- especially the new micro-processor technology -- and the changing needs of American higher education will require it.

APPENDIX A

CMI IN ECONOMICS

When the great depression deepened and extended into the 1930's and when the traditional economic strategies of cutting back production and waiting for things to level out brought no improvement, some economists began developing new economic strategies. Among those in the forefront of what was called the new economics was the English economic theorist John Maynard Keynes. In a now classic work, the General Theory of Employment, Interest, and Money, published in 1936, Keynes argued that

1. socialism or government ownership of the banks and means of production was the only way to end the depression quickly. 0
2. government spending ought to be increased in order to create new demands for goods and services. +
3. war or preparation for war was the only way to end the depression quickly. X
4. government spending ought to be cut back in order to reduce interest rates and make industrial expansion cheaper. Z

0 This answer is incorrect.
If Keynes was anything, he was a liberal, not a socialist. Though not a friend to the unbridled profit motive, Keynes never believed that any good would come from the state's running industry and trade. He considered the doctrine of state socialism to be obsolete -- state socialism was in his view a reaction to a 19th-

century political and economic environment which by his time had been changed out of all recognition. During the 1930s, he cherished the hope that his doctrines would wean the British Labour Party away from its commitment to state socialism. This did not happen.

Select another answer.

X This answer is incorrect.
Some writers have drawn this implication from Keynes' doctrines in order to discredit his ideas or to use caricatures of his ideas to discredit capitalism. If government spending was a way of increasing the amount of money in circulation and creating fresh demand, then war or preparation for war, which required government spending on a vast scale, ought to be the simplest and best of all possible ways to deal with unemployment and economic depression. This is what actually happened in France and the United States, but the advocacy of such a situation was not part of Keynesian theory. He did not believe that war was necessary for human progress. People could improve themselves by using their minds to build civilization, not to destroy it.

Select another answer.

Z This answer is incorrect.
From a Keynesian perspective, depressions were caused by shortages of consumption and demand. Interest rates, high or low, were irrelevant to the prospect of industrial expansion if consumption and demand had collapsed. Keynes argued for increased government spending -- not reduction of it -- as a means of increasing consumption and demand. According to Keynes, by increasing the amount of money in circulation, by coaxing savings into investment, and by providing people with money to spend, new demands would be created and new life breathed into the economy.

Select another answer.

+ Your answer is correct.
Keynes argued that the goal of economic activity was to achieve maximum volume of production and maximum employment. He argued that government spending, even deficit spending, would encourage production and employment by increasing demand. In Keynes' view, state socialism was obsolete. Limited government intervention in certain vital areas of the economy was the answer to economic progress. Public service jobs, occasional subsidies, and guaranteed loans and mortgages were all ways that a government could stimulate consumption and demand. While Keynes himself did not say that governments could spend their way out of depressions, this is how some popular writers oversimplified and misrepresented the message of Keynesian economics.

APPENDIX B

CMI IN HISTORY

By the middle of the 19th century, Romanticism as an intellectual movement had run its course. It was succeeded by a movement whose primary inspiration came from the natural sciences and technology. This new movement, which profoundly affected philosophy, painting, literature, music, and politics during the last half of the 19th century, was known as

1. Expressionsim. O
2. Impressionism. X
3. Realism. +
4. Symbolism. Z

O This answer is incorrect.
Expressionism found little inspiration in the natural sciences and technology. If it had a primary inspiration, it came from the behavioral science of psychology. Although there are some 19th-century writers and artists who can be described as Expressionists in some of their works -- Walt Whitman and August Strinberg, for example -- Expressionism did not become a major intellectual movement until the 1920s. It affected literature most strongly in Germany and painting most decisively in France. Expressionism was first and foremost subjective. Expressionists tried to communicate inner vision, feelings, emotions, intuition -- not ideas. They were not concerned about presenting appearances as they are, but with expressing essential and inner qualities. Thus, an Expressionist painter would not portray a mother and child; he would paint maternity. He would paint not Christ on the cross, but sorrow; not peasants eating potatoes, but poverty and degradation.

In general,Expressionistic writers depaired of life. Society, government, industry, religion, and humankind itself were presented as in a chaotic state. Disillusionment with the world as it is forced Expressionists to create a new world of psychological reaction and subjective experiences.

Select another answer.

X This answer is incorrect. Impressionism found little inspiration in the natural sciences and technology. If it had a primary inspiration, it came from the behavioral science of psychology.

Impressionist writers and artists have belonged to almost every age, but Impressionism as an intellectual movement appeared in the last two decades of the 19th century and mostly in France. Impressionists were concerned with attitudes, feelings, and emotions. They tried to record their impression of a scene, character, or incident. Impressionism shows us what we see after it has been interpreted by the mind and mood of an artist. The flash reveals the whole. The artist would reproduce the quality of a moment by appealing to our senses. Impressionist writers and artists would communicate moods rather than ideas. They would seek to produce emotions in their readers and viewers by a careful selection of appropriate detail. The objective of an Impressionist would not be to describe or depict fear. The Impressionist would try to have a reader or observer actually experience the emotion of fear while reading a passage or viewing a painting.

Select another answer.

Z This answer is incorrect. Symbolism found little inspiration in the natural sciences and technology. Symbolism is as old as art and literature, but as a movement, it became important in France during the 1880s. Symbolism in art and literature was really a narrower kind of Romanticism. Symbolists tried to present the essence of an idea or situation. Vagueness and even unintelligibility were not too great a price to pay if one could capture on paper the essence of a setting or mood. A Symbolist writer or painter presented what he saw in the mirror of his mind rather than what existed in nature. Content meant little to him. Symbolist writing was concerned intensely with emotions and often tinged with melancholy. Many Symbolists were pessimists, renunciants or resignationists. The aim of the Symbolists was not so much to communicate thoughts to others directly as to give them suggestions which would help them think or dream for themselves.

Select another answer.

Your answer is correct. The achievement of 19th-century science and technology was the wonder of the world. Nothing like it had happened since the age of Newton. General concepts, such as matter, energy, ether, organism, environment, and evolution, enabled scientists to organize enormous amounts of new information and greatly expand their knowledge of the physical world. The search for truth in the physical world inspired Realist movements in the worlds of art, literature, and politics. The interests of artists and writers focused on contemporary life. Realists marshalled facts and then stood aside. They tried to avoid emotional sympathies with their characters and subjects and strove mightily to keep personal feelings from intruding upon judgments. Realists were committed to facing life rather than running from it.

Realists in art and literature walked wherever humanity walked; they were often reformers and moralists. Realists in politics did whatever had to be done. They valued power over principle and rejected rhetoric for results.

EFFECTS OF LOCUS OF CONTROL AND PROVISION OF OVERVIEWS IN A COMPUTER-ASSISTED INSTRUCTION SEQUENCE

J.A. CAMPANIZZI
THE OHIO STATE UNIVERSITY

The effects of locus of control (learner control vs. program control of branching) and provision of overviews were investigated using a CAI sequence. Sixty college students were randomly assigned to four treatment conditions. Performance and response latency data were collected. Achievement gain was significantly greater under learner control than under program control. No significant main effects were found when using test response latency as the dependent measure. There was a significant difference between average response latencies of correct and incorrect responses on the posttest. A high negative correlation was observed between the time interval taken to make a response on the posttest and percentage of correct responses within the interval.

Development of a science of teaching and provision for the individual differences of learners are important priorities of education today. Computer-assisted instruction presents an excellent medium for realization of optimization of individualization in the instructional process. Computer-based educational technology also presents itself as a viable laboratory for the evaluation of the effectiveness of instructional strategies. Thus, important functions are served by the employment of computer technology in education.

There has been a limited amount of research utilizing computer-assisted instruction to evaluate the impact of instructional strategies upon the teaching-learning process. An important consideration for the educational researcher and the instructional developer is the identification of relevant strategies which promote optimum learning, while meeting the individual needs of the learner.

Stolurow (1968) suggests that computer systems provide an effective means for both empirical investigation and analysis of teaching processes. He presents computer-assisted instruction (CAI) as a "laboratory" or "theory machine", as well as being a way of implementing instruction. Computer-assisted instructional development requires relatively formalized techniques; controls are more easily provided than in most educational settings.

Computer-assisted instruction then presents a medium for analysis of the teaching-learning process as well as a means for instruction. If educators make wise use of this medium, then decisions regarding instruction could rely on empirical evidence rather than current trends or a "haphazard discard of old theory" (Stolurow, 1968, p. 68).

AREAS OF INVESTIGATION

Glaser (1976) states that research on instruction should attempt to evaluate the effects of strategies which maximize the outcomes of learning for individuals. Investigators such as Gilbert (1962) relate that some ways of organizing information may permit better memory retrieval than other ways, and thereby facilitate the learner's capacity to access information for thinking and problem solving.

Preinstructional organizers may occur in the form of pretests, behavioral objectives, overviews and advance organizers. The effectiveness of these forms of instructional organization needs to be investigated in a computer-assisted instructional setting where cost-effectiveness and efficiency are important considerations for determining instructional arrangement. A CAI environment where capacity for control of variables is optimum, will quite likely provide more reliable results than less formalized experimental settings.

Another area of instructional decision making, particularly in a CAI setting, involves determining the amount and type of control the learner is provided with during instruction. As the hardware capabilities of computer technology improve for interactive systems, it is becoming possible to provide the learner with more manipulative power during the instructional sequence.

Therefore, utilization of computer-assisted instruction involves making decisions regarding the amount of learner control versus amount of program control to allow for optimization of learning. O'Neal (1973) defends learner-controlled instruction (LCI) as a means for developing the self-directed learners required by modern society.

PURPOSE

The purpose of this study was to evaluate the effects of type of control (learner control vs. program control) and provision of overviews in a CAI sequence. The dependent variables utilized in the study were measures of achievement gain and response latency. An investigation was also made to determine if a significant correlation exists between response latency and performance.

The major questions of this study are: 1) What are the effects of type of control and provision of overviews on student on-line achievement and response latency? 2) Are there relationships which exist between response latency and performance?

METHOD

Design and Data Analysis

The experiment used a pretest-posttest 2x2 factorial design in which subjects were randomly assigned to one of four treatment conditions. Independent variables were: a) Locus of Control: Learner control or program control and b) Overviews or no overviews.

The treatment conditions developed were the following four CAI sequences:

LCOV - Learner control with overviews
LCNOV - Learner control with no overviews
PCOV - Program control with overviews
PCNOV - Program control with no overviews

Dependent variable measures taken in this study were: a) measures of achievement gain from pretest to posttest and b) measures of response latency. Data were subjected to analysis of variance.

Subjects

Subjects for this study consisted of sixty (60) college students enrolled in Health Education classes at The Ohio State University during the summer quarter, 1977. All subjects were randomly assigned to one of the four treatment groups, with a total of fifteen subjects per treatment. (See Table 1.) Fifteen participants (25 percent) had previous experience with CAI. The majority of participants (n=45, 75 percent) had no previous CAI experience.

	Learner Control	Program Control
Overviews	LCOV n = 15	PCOV n = 15
No Overviews	LCNOV n = 15	PCNOV n = 15

TABLE 1. THE FOUR TREATMENT CONDITIONS

Experimental Apparatus

Four instructional sequences were created by the investigator on the subject of venereal disease utilizing the Instructional Dialogue Authoring Facility (IDAF), a Hewlett-Packard CAI authoring language. The material presented in these instructional sequences was developed primarily from a text on concepts in human disease by Dr. Harmon C. Bickley, with permission.

The four instructional sequences consisted of 69 sections with adaptive branching employed. Therefore, the sequences were non-linear. The learner was first presented with an on-line nine item pretest. After the pretest the treatments LCOV (learner control with overviews) and PCOV (program control with overviews) were presented with an overview consisting of learning objectives adapted from Bickley (1974). These overviews consisted of short statements informing the learner of what principal concepts would be presented in the following instructional set. The overviews were presented before each of six instructional sets of the sequence. LCNOV (learner control with no overviews) and PCNOV (program control with no overviews) treatments did not receive the overviews.

After each instructional set, the learner received a formative evaluation instrument consisting of two or three multiple choice questions over the previous instruction. Subjects responded by typing the letter (A,B,C,D or E) of the correct item and depressing the RETURN key. Feedback was given as to the correctness or incorrectness of the response. A criterion was established for passing each formative evaluation instrument. If the criterion was not met, the subjects in the learner control treatments (LCOV and LCNOV) were advised that they had not met the criterion of acceptable performance, and given an option to branch to a set consisting of a review of the previous instruction. Subjects not achieving criterion in the program control treatments (PCOV and PCNOV) were automatically branched to the review set by the program, without receiving an option to branch.

These procedures were repeated throughout the six instruction sets. Upon completion of these sets, the learner was presented with an on-line nine item posttest which ended the sequence.

Pretest and Posttest

Pretest and posttest items were taken from three parallel forms of tests accompanying the section on venereal disease of Dr. Bickley's text. The tests were developed in the Department of Pathology, at the University of Iowa College of Medicine. The pretest and posttest were of multiple choice format, with five responses to each test item. The learner responded to the test item by selecting the letter A, B, C, D or E, then depressing the RETURN key.

Data Collection

Response latency data was measured and collected for every section of the sequence by the internal timer of the Hewlett-Packard 2000F computer system. The latency of each subject's response was written onto a disc file. Response data for each subject was collected from the listing of DECwriter hardcopy terminals printing at 30 characters per second, which served as the medium of presentation of the sequence.

Procedure

Upon entering the room, subjects completed a demographic data form and a consent form. A card containing the name of the treatment and a subject number was drawn from a shuffled deck to determine the treatment assignment. Subjects were then instructed to be seated at the terminal, and given a printed paragraph to read explaining the procedure for correcting a typing error. The investigator then signed each participant on to the computer terminal for the instructional session. After completion of the session, the subject listing was saved for data collection purposes, and subject response latency data was transferred to a permanent holding file.

RESULTS

Effects of Treatments

Learner control appeared to influence achievemtn gain while overviews appeared to not have a significant effect on achievement gain. Data were subjected to a two-way analysis of variance. The results indicate a significant effect for type of control ($F=4.31$, $p < .05$, [dfl, 56]). The learner control treatment group produced a mean gain of 4.73 as opposed to a gain of 3.77 for the program control group. No significant main effects for treatments were found when using test response latency as the dependent measure.

No significant differences were found among treatment groups in the number of times the review sequences could have been presented; however, the learner control group actually elected to receive the review sequences significantly fewer times than the program presented these sequences to the program control group ($F=6.4$, $p < .05$, [df 1, 56]). (See Table 2.)

Response Latency and Performance

Average test response latency (ATRL) is the sum of the response latencies on a test divided by the number of items on the test. Average test response latency change (ATRLC) is obtained by subtracting the average test response latency (ATRL) on the pretest from the ATRL on the posttest.

ATRL data were collected for all subjects in all treatments both on the pretests and posttests. Data were analyzed using type of test as the independent variable. For ATRL derived from correct and incorrect responses, latencies were longer on the pretest than on the posttest ($F=47.76$, $p < .001$, [df 1, 118]). For ATRL obtained from correct responses only, group latencies also were longer on the pretest than on the posttest ($F=52.72$, $p < .001$, [df 1, 106]). Data were also analyzed obtaining ATRL from incorrect items with no significant effects observed. Figure 1 presents a graphic representation of group means.

ATRL data were collected for all subjects, first for correct responses only, then for incorrect responses only, without regard to treatment groups. On the pretest, no significant effects were found comparing items responded to correctly and items responded to incorrectly. On the posttest, correct items had a shorter mean response latency than incorrect items ($F=16.27$, $p < .001$, [df 1, 102]). Group means are presented graphically in Figure 2.

Response latencies were collected for each of the nine posttest items of each subject in the study. Therefore, latencies were obtained for 540 items. Each item was assigned to one of eight time slots. These time slots reflected the subject's response latency to a posttest item. Items within each time slot were then evaluated as to correctness or incorrectness of response, and the percentage of correct items within a given time slot was calculated. Figure 3 provides a plot of the relationship between variables x (response latency - number of seconds to respond on posttest items) and y (percentage of correct items). Results show a high negative correlation of -0.95844 (Pearson's r) at the 0.0009 significance level. As response latency to test items rose, the percentage of correct responses to items went down.

DISCUSSION

One is led to speculate as to the reasons for the significantly greater achievement

TABLE 2
ANALYSIS RELATED TO THE BRANCHING CONDITION

DEPENDENT VARIABLE	LOCUS OF CONTROL (LC, PC)	OVERVIEWS (OV, NOV)
RC_1	NS	NS
RC_2	LC < PC	NS

RC_1	number of times criterion not met
RC_2	number of times recap given
LC	learner control
PC	program control
OV	provision of overviews
NOV	no provision of overviews
NS	not significant

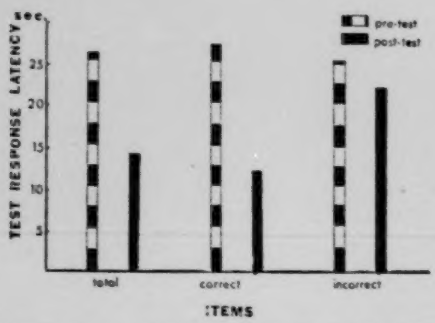


FIGURE 1. GRAPH OF TEST RESPONSE LATENCY BY ITEMS

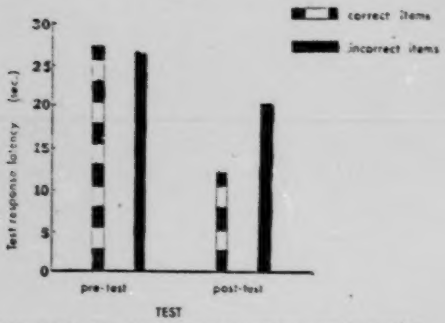


FIGURE 2. GRAPH OF TEST RESPONSE LATENCY BY TEST

gain under learner control as opposed to program control. Judd (1975) suggested two reasons for his support of learner controlled CAI. He implied that instructional control provided the learner is less aversive to the individual than instruction administered under program control. Secondly, Judd optimistically states that the student is "sufficiently aware of his state of learning to make his own instructional decisions." (p. 2)

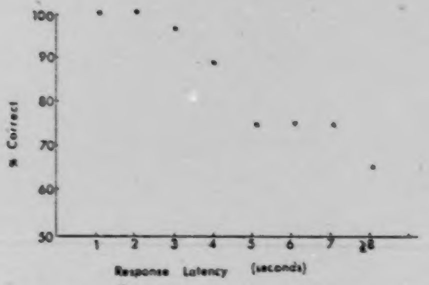


FIGURE 3. GRAPH OF RESPONSE LATENCY

It is interesting to note that there was no significant difference in the amount of times the evaluation criteria were not met by the subjects in the learner control and program control groups. That is, both of these treatment groups potentially could receive the review a relatively equal number of times. However, the learner control group actually received the review significantly fewer times than the program control group.

This observation may indicate that the learner is quite aware of his learning state duirng instruction. Results were obtained even with the builtin facility advising the learner to take the review sequence when the criterion was not attained. Given the opportunity to actively participate in decision-making, the individual may not always follow the advice, but will choose paths which will ultimately result in achievement.

One may suggest that corrective feedback presented during the formative evaluation provided the learner with enough information regarding his acquisition errors; the learner, then realizing his mistakes, did not feel the need to review.

The outcomes of this investigation would support Judd's second contention. It may also be argued that being given an active role in the instructional activity contributes toward making the material more meaningful for the learner, hence less aversive than material in which the individual maintains a more passive role. However, accurate measures of state of anxiety would need to be employed to obtain a clear indication of the validity of this argument.

The study resulted in no significant effects for objectives. The overviews presented in the sequence were not of highly specific nature, but served as general organizational indicators of forthcoming material. This type of overview did not appear to contribute to significant achievement gain. Pretests may have had a preorganizational effect which made the additional effect of overviews not discernible.

The investigation also explored suggestions in Judd and Glaser (1969) that response latency may be a sensitive measure of strength of learning. The findings seem to require an in-depth study of this variable.

Merrill (1974) found that while objectives had no effect on achievement, they did significantly effect test-item response latency. Hence, Merrill postulated that response latency may be a more sensitive indicator of the internal state of the learner than performance measures. The results of this study do not support his argument. No significant effects for treatments were found by using either response latency on the posttest as the dependent variable or response latency change as the indicator.

As this study indicates, however, it is important to look at response latency when it is related to the correctness or incorrectness of test items.

Results of this investigation demonstrate that after the acquisition phase of learning, as indicated on the response latencies of the posttest, there is a significant difference between response latencies of correct and incorrect items. Average response latencies of correct items are significantly shorter than response latencies of incorrect items. The response latencies measured on the pretest do not support this statement. Response latency measures may indeed reflect internal states of the learner and serve as an indicator of how well information is being stored and retrieved by the learner.

The negative correlation between percentage of correct responses to items and response latency needs to be addressed. As response latency rose, percentage of correct responses decreased. If one examines the plot of this correlation, an observable change appears to occur around four seconds. Before four seconds percentage of correct items appears quite high, becoming much lower after four seconds.

With regard to average test response latency, average latencies on the pretest appear to be significantly greater than average latencies on the posttest. This observation seems to indicate two things. It tends to provide affirmative support to Judd's (1973) suggestion in paired-associate studies that latency changes over the course of learning. It also suggests that the learner, by spending more time on the pretest, has demonstrated willingness to approach the instructional environment with a concern for performing well. No external performance incentives such as grades, monetary payoffs, etc., were provided in this study. Acquisition of information may have been an internalized incentive.

REFERENCES

Gilbert, T.F. Mathetics: The Technology of Instruction, Journal of Mathetics, Vol. 1, pp. 7-73, 1962.

Glaser, Robert, Components of a Psychology of Instruction: Toward a Science of Design. Review of Educational Research, Vol. 46, No. 1, pp. 1-24, Winter, 1976.

Judd, Wilson, et.al. "Individual Differences in Learner Controlled CAI". Paper presented at the Annual Meeting of the American Educational Research Association, Report IR 001 951, Washington, D.C., April 1975.

Judd, Wilson A. and Robert Glaser. Response Latency as a Function of Training Method, Information Level, Acquisition and Overlearning. Journal of Educational Psychology Monograph, Vol. 60, No. 4, Part 2, August 1969.

Judd, Wilson A. Glaser, Ribert and Rosenthal, Daniel, J.A. "Response Latency As a Correlate of Individual Differences in Retention", Paper presented at the American Educational Research Association Annual Meeting, New Orleans, Report EM 010 931, March 25, 1973.

Merrill, Paul F. Effects of the Availability of Objectives and/or Rules on the Learning Process. Journal of Educational Psychology, Vol. 66, No. 4, pp. 534-539, 1974.

O'Neal, Fred, Learner Control of Instruction: Requirements and Potentials. Paper presented at the ADCIS Conference, ICUE TEchnical Report #2, Ann Arbor, Michigan, August 1973.

Stolurow, Lawrence M. Computer-Assisted Instruction, American Data Processing, Inc., Detroit, 1968.

EIGHTEEN HANDS ARE BETTER THAN TWO -- THE MULTIPLEXER APPROACH TO COMPUTER ASSISTED INSTRUCTION

Stanley A. Crane
Project MASTER, Garden City High School
Garden City, Kansas 67846

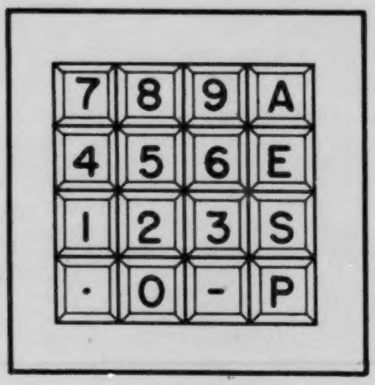
Computer Aided Instruction has always been plagued by its high cost. Therefore, it has not had a great impact on large numbers of students. To solve this problem, Project MASTER[1] has multiplexed eighteen alpha-numeric keypads in place of two terminals. Presently, mathematics, science, remediation of basic skills, and English programs make use of the calculator-like keypads. Project MASTER now serves the needs of up to 700 students per week. The impact on the educational process is evidenced by the doubling of Garden City High School's mathematics enrollment during Project MASTER's three years.

People involved with computer aided instruction have always been confronted with two problems: How can cost-effective equipment be found, and what can be done that is instructionally valuable with this expensive hardware?

Several computer aided instruction systems have been remarkably effective, yet the cost prohibits either the use by large numbers of students, or the use in relatively small school systems.

In planning an approach to computer aided instruction, every computer systems manager must keep in mind that the source of his funding is a group of people that is money conscious. One possible alternative to the traditional $1000 to $4000 computer terminal is the use of keypads or "mini-terminals". At MASTER, these keypads have been linked to a computer terminal and used to provide a means of input to the computer. Output is via the terminal and is displayed on a large video monitor. The keypads were engineered and built especially for Project MASTER. There are eighteen such keypads, one cluster of eight and one cluster of ten. The group of eight together with a multiplexer cost $5500. Price is not the only factor that helps to make the keypads more economical. Each set of keypads acts as a single device, requiring only one program in core for each set. Aside from the direct savings in core, a smaller computer, not capable of handling eighteen terminals, can use much simpler software and be extremely functional. The programs that drive the keypads operate in less than 10K bytes of core, per set of keypads. A comparable setup using $1000 terminals, running programs of 5K bytes would cost $18,000 and would also demand the cost of an additional 70K bytes of core. Using a multiplexer, eighteen students are now accomodated where previously there were only two.

The keypads in use at Garden City High School have only sixteen keys, a major factor in the reduced cost of the keypads. Fewer keys have also helped by lowering the frustration level of the students; even lower elementary students can quickly find a particular key.



The keypads have the characters 0 thru 9, . (decimal point), - (minus), E (exponent) S (student number), A (answers), P (process or send).

---

[1] Project MASTER was made possible by a grant from Title IV-C and is presently in its third and final year of Federal funding.

Before presenting Project MASTER's particular systems, it is imperative to point out that primarily the instructional component in computer assisted instruction should be the responsibility of the classroom teachers and curriculum personnel. Computer programmers should not develop a system in total isolation and expect it to be adopted and then complement teachers' methods. This generates ill feelings and demonstrates a lack of confidence in the teachers. Teachers educated about computer capabilities and new developments in hardware can help programmers to develop invaluable insight into the instructional process. Teachers should be encouraged so that any good idea will be utilized and nurtured with the cooperation of all those affected. By following this criterion, the final product is more readily useable in the classroom. In addition, enlisting teachers' help in the developmental process pays many dividends in terms of their understanding should bugs develop in a program.

At Project MASTER, three types of supplements to the classroom routine are available. The first deals with problems which have numerical solutions, such as most mathematics and science problems. This portion of MASTER has been utilized for the past three years.

Random problem sets, tailored to a particular student, are printed on a line printer after the teacher and student provide the following information:

Topic:
supplied by the teacher
Career:
given by the student
Difficulty:
provided by the teacher or student
Number of problems:
determined by the teacher.

The student types the solution to a problem on one of the keypads. The computer responds on the monitor with an appropriate congratulatory or error message. A typical input-output cycle might produce the following:

```
PAD 6, RICHARD, PROBLEM 3
CORRECT - CALL THE PRESIDENT - TELL
HIM TO DECLARE A NATIONAL HOLIDAY!
PAD 3, JOANNE, PROBLEM 2
RIGHT! CORRECT! FANTASTIC! OUTA
SIGHT!!
PAD 7, JEAN, PROBLEM 2
UNLIKE A STUNT PILOT, YOU CAN'T
PERFORM YOUR OPERATION UPSIDE DOWN
PAD 8, ALLEN, PROBLEM 6
EXCELLENT! IF I HAD LIPS I'D KISS
YOU!
PAD 1, AUDREY, PROBLEM 4
I HAVEN'T MISPLACED A DECIMAL LIKE
THAT SINCE I WAS KNEE HIGH TO A
CALCULATOR.
```

The student may go on to another problem or rework the same problem if his answer was incorrect. At the end of the day, teachers receive reports analyzing each student's work. Upon request, a summary reflecting a complete history for an individual student is available.

The second supplement is used for multiple choice questions. This program has been a part of MASTER for one year. It is presently used for English exercises in grammar, vocabulary, spelling, and proof reading. An example of a spelling problem looks like:

Which of these words is spelled correctly?
1) reign
2) riegn

A typical set of questions includes three groups of problems. The student works the medium difficulty group and inputs his answers in the same manner as described for the mathematics and science problems. Based on his performance, he is directed by the computer to work on either the easier or the more difficult group of problems. Whether a student works on identifying pronouns, calculating the heat in a chemical reaction, or figures the displacement of an engine, the computer assists by evaluating his work and directing him toward a meaningful learning experience. There are some minor differences between these first two systems, but for the most part they function similarly in providing immediate responses to individualized sets of problems.

The third type of instruction, implemented two years ago, uses the keypads in a drill and practice routine for remediation and mastery of basic skills.

The teacher specifies what operation(s) will be used, upper and lower bounds on the operands (including decimal point location), and what error range will be acceptable. A .1% range is used for exact computation whereas a 20% range might be used for practice on estimations. The computer generates the problems and prints them on the screen. This has been used regularly by third grade through high school students. Every student in a group must type in the correct answer to the problem before the computer will advance to the next problem. This system also provides daily and summary reports of student work.

With these three systems Project MASTER has augmented the education of 950 of Garden City High School's 1,100 students this year. Weekly usage has peaked at nearly 700 students. The computer has been used by virtually all of Garden City High School's seniors, making it one of the few "special" programs that has impacted the education process as a whole. In addition, the summary reports provided by these systems could prove to be invaluable in establishing a competency based educational system or furnishing exact data for accountability. The greatest evidence of Project MASTER's influence, however, has been in the area of mathematics enrollment. In 1975, when MASTER began, the percentage of students enrolled in mathematics classes was close to an all time low, 38%. This year, 76% of the students are taking a math class, 10% higher than the previous record of 65% in 1966. In 1966, two years of mathematics was required, presently only one and one-half years are required. Project MASTER has been the only change in the mathematics curriculum.

Based on the experience gained through Project MASTER, the multiplexer approach to Computer Aided Instruction offers a realistic and effective method of integrating a computer into a curriculum.

ON THE CONSTRUCTION AND USE OF COMPUTERIZED SIMULATIONS

J. Richard Dennis
Department of Secondary Education
University of Illinois, Urbana

One role for computers in education that appears to justify the resource demand is instruction via simulation. This application accounts for a significant portion of the literature of educational computing and yet remains an activity about which relatively little is known. In part this is due to the state of infancy prevailing throughout educational computing. To date, efforts have been directed largely at producing demonstrations of several instructional styles. Little attention has been paid to inherent features of these styles. Yet taxonomies or definitions are essential to the building of a research literature. In this paper we address the questions: What is simulation? We present the defining characteristics for the activities called simulation, together with a taxonomy for distinguishing these activities from among themselves. The subsets of this taxonomy are determined by the learning objectives apparently shared by the members of the subset.

Simulations have become an integral part of educational and training programs in several areas of human activity. Medical schools use simulations for instruction, for practice, and for evaluation in the development of diagnostic and other decision-making skills required of physicians. Business management curricula frequently include simulations as an integral part of activities centering on problem solving and solution implementation. Training for other decision-making or performance centered occupations (e.g., airline pilots) includes simulation of on-the-job activities for initial instruction, for evaluation, and for retraining. And several have argued for the effectiveness of simulations in the teaching of the structure and operation of social systems.

There is an increasing incidence of simulation usage partly due to successes at computerizing such activities. At the same time there is an ongoing debate about the benefits, or lack thereof, of this instructional style. Proponents list increased motivation, realization of the role of good communication skills, forced decision-making (often based upon incomplete data), learning to accept results of and responsibility for previous decisions regardless of their quality, and the role of teamwork and responsibility as reasons for, or outcomes of, the use of simulation. Those who question the value of such activity point to the lack of data on the educational value of such experiences and to the cost in time and financial resources usually required to produce and to utilize simulations. Both sides agree on the increased enthusiasm and motivation usually observed among learners during use of simulations.

More detailed analysis of real or hypothesized outcomes of simulation will be presented later in this work. For now, the picture presented above suffices to illustrate the inconclusive nature of the debate on simulation as a learning medium. A major contributing factor to this uncertainty is the fact that different authors use the term "simulation" to refer to a wide variety of activities, not always discernable as similar. The same uncertainty frequently is present in discussions of simulation games, and sometimes in discussions of games.

Such situations are not new to the field of educational application of technology, nor are they new to more broadly conceived research topics in education. During the fifties and sixties, for example, many researchers turned their attention to "discovery learning." This "discovery" literature presents another situation in which lack of common, or even carefully articulated definitions has injected an element of inconclustion, if not confusion to the literature.

In this paper we first shall address simulation from a definitional perspective, and then present a taxonomy for accounting for differences among members of the resulting class of activities. Our starting points in this exercise are a wide array of instances presented in the literature under the title "simulation." These include flight simulations, stock company simulations, simulated laboratories, simulations of medical diagnosis, machine procedure simulations, competition simulations, population or other ecological system simulations, etc.

What is a Simulation?

In an attempt to clarify terminology, we first imagine two sets, one whose members we shall call simulations, and the other whose members we shall call games. Out interest here is in the first of these sets.

A simulation is a representation of an aspect or phenomenon of one's universe. The behavior of the phenomenon is determined by a set of principles or properties. The representation is designed to make experiences or interactions with it to be in some way congruent to corresponding experiences or interactions with the phenomenon itself. But the experiences with the representation are more convenient and more freely manipulable than their counterparts of the phenomenon. In this representation one is confronted by a sequence of surrogates to problematic aspects of the phenomenon. These encounters are ones that require inquiry, decision, or action. And each of these, in turn, is accompanied by feedback in the form of knowledge of consequences or altered states of the environment of the phenomenon. The participants goal in executing the simulation is to achieve an ability to control or to understand the variables and interactions of the phenomenon.

For contrast and comparison we need to define members of the other set, the games. A more elaborate discussion may be found in a work by Fletcher (1971, 429-430). Briefly, most definitions of games include: (1) a set of more than one decision making entities called players, each with a capacity to act; (2) a conflict or competition among the players; (3) a set of procedural rules; (4) a set of specified outcomes; (5) each player has a pattern of preferences among the outcomes; and (6) an information system. The goal of each participant of a game is to beat the rules or to win.

From these definitions we can see two features which can serve to distinguish "pure" games from "pure" simulations, and to explain the meaning of the term simulation game. A game involves a competition among participants. A simulation need not have a competition nor need it have more than one participant. A simulation is a representation of an aspect of our experiential universe, while a game need have no counterpart in reality. In a game, permitted actions are defined by rules of play, which serve to structure and to explain the competition. But in a simulation, permitted actions are defined in terms of properties of the phenomenon and relationships among its variables.

So what we have are two distinct, but not disjoint sets--the simulations and the games. Their common elements are called simulation games. A simulation game is characterized both by a representation of a phenomenon and by a competition. The rules of the competition and the challenges presented by the problematic aspects of the phenomenon overlap. Winning the game takes on the form of a "best" achievement of control or understanding of the represented phenomenon.

A Taxonomy For Simulations

The preceding discussion serves to define simulations from a gross point of view. However, closer examination of a variety of instances reveals variance among them on several attributes. One of these varying features is the learning goal or objective to which a particular simulation seems to be directed. The subsets emerging from an analysis of these differences are as follows:

1. Replicable Performance Simulations
   Those whose learning outcome is the replication of a specifiable performance. The performance may be physical, intellectual, or both. The necessary and sufficient data, decisions, actions, and consequences are the ingredients of the simulation.
2. Information Retrieval Simulations
   Those whose learning outcome is information in the form of facts, principles, or understandings, all of which are specifiable and/or quantifiable. Although performances may be implied by such information, achievement of a level of mastery of the performance is not a goal.
3. Encounter Simulations
   Those whose learning outcome is the "experiencing" of a situation. One is expected to become aware of possibilities or probabilities of the situation. The learning goal is awareness of variability among consequences, and procedures for assessing the magnitude of this variability.

In the next sections we describe in more detail, the characteristics of these three types of simulation.

Replicable Performance Simulations

All instances of this category have as their goal an expected replication of a series of behaviors in an optimal sequence. The associated real performance is task oriented, the task is definable. Examples of this category are diagnostic and/or prescription simulations from several fields, procedure simulations from several vocations or from laboratory settings, etc.

The real phenomena associated with the members of this class have a goal to reach, but reaching the goal may or may not involve problem solving. The phenomena have two components, one a procedural logic, the other a procedural execution. The role of a simulation in this category exclusively is to gain proficiency with the procedural logic. In so doing, however, opportunities are presented in the simulation that usually are not present in reality. One has the opportunity deliberately to explore and to gain knowledge of the consequences of other than optimal actions and decisions.

An important characterizing feature of these simulations is the relationship between the learnings of the simulation and optimal performance in the real setting. Behaviors or knowledge mastered during simulation have associated with them a guaranteed applicability in the real setting. This is not to say that transfer of learning is guaranteed, but rather that, when transfer occurs, the effectiveness of the performance learned is guaranteed. The nature of the phenomena associated with these simulations accounts in great part for such guarantees. Procedures are definable, consequences of actions are determinable and there is an association of certainty or known expectation between each action or decision and its consequences.

Information Retrieval Simulations

Simulations in this category are those whose goal is the gaining of information. No clear tasks or performances are obvious in the setting. The associated phenomenon is a process which operates under a set of principles or laws. In the absence of clear tasks or goals, there are no optimal solutions or strategies involved. Examples of this style simulation are those presenting a variety of physical or social processes such as genetic reproduction, economics, ecological systems, and other scientifically based phenomena.

The phenomena underlying these simulations are well understood, and the interactions of their components or variables are defined and understood. The role of the simulation in these situations is information transmitted in several forms. In the lowest form, the information takes the form of facts about states of the phenomena. At more sophisticated levels the information is in the form of principles of cause and effect in the phenomena, or of principles for achievement of steady states.

In Information Retrieval simulations the participant may have very vague goals, or may be pursuing the effects of definitely specified experiments on the surrogate environment. To do so, the participant makes changes in variables in the phenomenon and then seeks data on the state changes accruing from such alterations. Skills are not an outcome of the simulation except, possibly, in the remote realm of inquiry skills or experimental skills. No performance directly is implied by participation in the simulation or by the resulting learnings. Learning is defined by the extent to which the designing of experiments followed by observation of consequences can be changed into an ability to define experiments and predict their consequences prior to execution of the simulation. An intermediate stage in this learning continuum is the ability to design an experiment and then to explain the results once they are observed.

Encounter Simulations

Simulations of this variety are concerned with phenomena that neither are completely definable nor predictable. The purposes of such simulations is to gain experience in achieving a goal, or effectively functioning in a setting in which there exists no clearly optimal procedures or principles. This is a result of the nature of the phenomena involved. These phenomena have weak or irregular relationships between actions and results. Examples of this category are simulations of business competitions, of discipline episodes in teaching, of locating missing persons in unfamiliar enviornments, of a teachers first year of experience toward achieving tenure, and the wide variety of social role playing activities.

The phenomena underlying these simulations usually are not well understood. The variables influencing behavior of the phenomenon are not all identified and the interactions among known variables are influenced by sometimes unpredictable, chance events. The role of simulation in this case is to provide opportunities to gain familiarity with possibilities and with the nature of the chance influences. There are few cause and effect relationships. Through the simulation one tries to derive incomplete understandings of the varying relationships between events or actions and the array of likely or unlikely results. No direct transfer of experience is guaranteed. Neither is the applicability of any learnings guaranteed to similar settings.

Features of A Simulation

A simulation can exist for any phenomenon characterized by a problem to be solved, a goal to be reached, or an environment to be understood. These phenomena usually are complex in nature, possessing a multitude of variables and interactions among them. Many methods may lead to a solution, to obtaining the goal, or to achieving understanding. A small number of these methods may in some way be designated as "better" than the others.

A simulation derived from such a phenomenon attempts to focus on a bounded portion of the environment. The environment is restricted to ignore selected aspects. The simulation focuses ones attention on a central "field" in much the same way that a photographer focuses vision on the central subject matter of a visual life by avoiding complications surrounding the principal subject matter. A limited amount of the more pertinent complications may be included to gain perspective or realism. The important aspects of the simulation are that all opportunities for data collection, for decision, and for action present in the real environment are reflected in the surrogate.

A simulation is initiated by an opening scene in which the setting is described. Mentioned in the opening scene are those aspects of the setting which readily are available in reality. A balance is required between overwhelming with irrelevant data and overly digesting so as blatantly to cue an obvious solution or understanding. Some selected irrelevancies may be included to achieve realism, or to motivate participation, but every attempt is made to avoid deliberately misleading the participant. Also, in the opening scene, the participant's role is spelled out, a task is stated, and any limitations in time, facilities, or available assistance are revealed.

The heart of a simulation consists of a set of ordered pairs whose first components are options (choices) relative to the setting and whose second components are the responses (consequences) of instituting that option. McGuire, et al (8) have presented four classes of option pairs usually present in situations being simulated:

1. Data Gathering Options--methods of access to accumulated information some of which is needed to complete the task.
2. Management Options--decisions or actions that may alter the environment.
3. Consultation Options--seeking authoritative help on data collection and interpretation, on making management decisions, etc.
4. Valuing Options--expressions of personal attitude or feeling about the problem or setting. These are derived from an assessment of the possible value positions in the situation.

Embedded in a simulation is one or more prescriptions of "correct" action. At each point in time, however, one must be permitted access to all those options that would be accessible in reality. Within the resulting network then, a task completion method is characterized by a sequence of choices representing an optimal path (or sequence) from among the set of all options. Of course, a variety of paths may be possible, including less optimal ones, and ones resuling in abortions based upon misconceptions. The later option paths are difficult to build in, since they require determination of the misconceptions likely to be present in the context. So, each option in a simulation is of one of five types depending upon the resulting progress toward completing the task. These types are:

1. Necessary Option--required for successful completion.
2. Expediting Option--not necessary, but will facilitate completion by revealing or implying necessary actions.
3. Neutral Option--non-contributing to task completion.
4. Delaying Option--impedes progress and may imply or reveal an impending abortion.
5. Aborting Option--renders completion impossible.

The response parts of the option pairs inform of changes in conditions in the simulation environment. Such responses usually are not evaluative, not corrective, and not explanatory. This is the way the real situation would be. In some cases, even the data on resulting changes would need to be requested because in reality it would not immediately be obvious following the action.

Each simulation has a single or series of strategy initiation sections representing the major decision points at which choices among the available options is made. Visitation of a strategy initiation point occurs each time the consequences of a previous strategy has been played to completion.

Finally, a simulation has a collection of transition messages which facilitate the participant's execution of the simulation. These are independent of successful completion of the task of the simulation. Rather they serve to inform the participant of the actions to take to keep the simulation progressing. These are particularly important in computer-based simulations to inform the user of mechanical aspects such as keyboard functioning.

Summary

This paper has presented an analysis of the characteristics of and types of simulations. A major purpose for identifying characteristics is to organize and to simplify the task of constructing new simulations. A major purpose for establishing a taxonomy of types of simulations is as a basis for the design of research into the effects of simulations in instructional settings.

A detailed presentation of research findings and hypotheses on learnings derived from simulation experience is the subject of a much more lengthy review. Briefly, several reports, from a variety of learning contexts, attest to signficantly increased attitudes through the use of simulation. A small number have hypothesized cognitive outcomes of a wide variety, some representing learnings apparently not achievable by other methods. A still smaller number have presented preliminary data that significantly different cognitive outcomes have been identified in selected instances of simulation use. There remain at this point many unanswered questions about the learning effects and usefulness of this exciting, controversial medium. But, answering these questions partly is dependent upon more individual experimentation with the instructional style coupled with systematic documentation of methods and results.

Bibliography

1. Abt, C. C.; "Games for Learning" in Boocock and Schild (eds.) Simulation Games in Learning, Sage Publications, 1968, 65-84.
2. Bewley, W., D. Holznagel and D. Kalssen, "Toward a Cognitive--Developmental Rationale for the Instructional Use of Computer Simulations," in Computers in Education, O. Lecarne and R. Lewis (eds.), IFIP, North-Holland Publishing Co., 1975.
3. Boocock, S. S. and Schild, E. O. (eds.) Simulation Games in Learning, Beverley Hills, Sage Publications, 1968.
4. Chartier, Myron R., "Learning Effect: An Experimental Study of A Simulation Game and Instrumented Discussion," Simulation and Games, Volume 3, Number 2, June, 1972, 203-218.
5. Cherryholmes, C. H., "Some Current Research of Effectiveness on Education Simulations: Implications for Alternate Strategies," American Behavioral Scientist, Volume 10, Number 2, 1966, 4-7.
6. Fletcher, J. L., "The Effectiveness of Simulation Games as Learning Environments," Simulation and Games, Volume 2, Number 4, 1971, 425-454.
7. Harry, L., "Using Simulation Games in the Classroom," John Hopkins University Center for the Study of Social Organization of Schools, 1969.
8. McGuire, C., L. Solomon, and P. G. Bashook, Construction and Use of Written Simulations, The Psychological Corporation, 1977.
9. McKenney, J. L., "An Evaluation of a Business Game in an MBA Curriculum," Journal of Business, Volume 35, July, 1962, 278-286.
10. McKenney, J. L., Simulation Gaming for Management Development, Cambridge, Massachussetts, Harvard University Press, 1967.
11. McKenney, James L. and Peter G. W. Keen, "How Managers Minds Work", Harvard Business Review, Vol. 52, No. 3.
12. Piel, E. J., "Evaluation of Huntington II Project on Computer Simulation," J. Education Tech. Systems, Vol. 3, No. 4., Spring, 1975.
13. Pollack, Harvey, "Computer Simulated Laboratory (CSL): Rationale and Approaches to the Structuring of CSL Exercises" Educ. Tech., March, 1976.
14. Robinson, J. A. et al, "Teaching with Internation Simulation and Case Studies," American Political Science Review, Volume 60, March, 1966, 53-65.
15. Twelker, Paul A., "Some Reflections on Instructional Simulations and Gaming," Simulations and Games, Volume 3, Number 2, June, 1972, 147-154.

USING THE CALCULATOR ASSISTED GAME ECONOMY (CAGE): A SIXTH GRADE EXPERIMENT

Herbert W. Fraser
Earlham College

A modest proposal to intorduce a little scientific rigor into the teaching of secondary school social studies courses. How this can be done in units devoted to the study of economics as required for teaching current problems and history courses by using the framework of the conditional prediction is shown. How CAGE can assist the teacher to learn the basic economic theory of production, income distribution and growth by allowing him to simulate a developing economy which his students will later simulate in class is illustrated by an experiment with sixth graders.

Three interrelated deficiencies in high school education go far to guarantee that the majority of students will be poorly equipped to understand the world as it is (let alone as it should be): First, only a small minority of students learn anything about the methods of science and, even then, only as applied to the natural sciences. Second, even a smaller minority learn how to use modern computational equipment. Studies dealing with the interaction of even a small number of variables are, therefore, denied to most students so that either they learn simplified unicausal explanations for complex phenomena or that everything is so complicated that the outcomes are all a matter of opinion. No guidance is given as to what constitutes better or worse opinion. Third, even as it is becoming more and more obvious that all of the major social and political problems in the United States, as well as in the world at large, are heavily loaded with economic considerations, few students are given any exposure to economic analysis either in formal courses or as part of their training in current problems, history or government courses.

The method of instruction to be described in this paper has been designed specifically to overcome these deficiencies starting as early as in the sixth grade. It employs several "scale model economies" beginning with a very simple one-commodity, two-factor world ahd progressing in sophistication to a three-commodity, three-factor, two-country world. Each model specifies technology and maximum potential resource availability but allows students to specify their own preferences and institutional arrangements. These model economies permit students individually, in small groups or in classes of practically any size to learn basic economic theory by experiencing and participating in a series of interrelated simulation games that see the economy they help shape evolve from feudal beginnings through a single-crop less developed country and finally into a capital-using, multiproduct economy engaged in foreign trade.

I. Three Uses for CAGE

In the simplest of these simulations (designed originally for use in the 9th grade) computational equipment is not required although a magnetic card programmable calculator can be used to good advantage in three distinct ways: First, a teacher who is unfamiliar with economic theory can simulate, by using CAGE as a game of solitaire, the hypothetical economy his or her students will later simulate in the classroom. Each evolution of the economy can be explored before the students are exposed to it while, using the teachers manual, the teacher can teach himself the required economic theory. The programs are designed to let the player approximate by trial and error (as in a real market and as the students later will do) the solution to the basic questions all economies must answer: What? How? and For Whom? to produce.

The second way CAGE can be used is by groups of teachers, students, or anyone else, as a calculator-assisted board game in which, as in real economies but unlike in most games, players maximize satisfaction as they define it by selecting weights to be attached to five arguments in their utility functions: present consumption, future income, leisure, income equality

138

and an index of xenophobia. By giving this choice to players the game becomes ideologically neutral.

The third use is with simulations in class. If the class is too small to produce competitive results, CAGE provides any number of dummy players required. In large classes students key in to the calculator all of their transactions to record them on magnetic cards. At the close of the simulation exercise the cards are gathered and run through the calculator which credits and debits the various participant's income and capital accounts and prints them out for the students to check. National Income accounts, and data on employment and earnings in various occupations are calculated and printed--the basic data that theory is called on to explain.

II. Conditional Predictions, Facts and Values in Economies

All of the simulations have one basic teaching objective in mind: to teach students how to make conditional predictions of the type "If A then B in time C" about their hypothetical economy. Making conditional predictions is, of course, the objective of all sciences so that what students learn about the process in economics can be used to let them find out what the natural sciences are trying to do.

In economic theory the conditional predictions have to do with how scarcity of resources, given technology, peoples' preferences and institutional arrangements (the conditions) force people to choose What goods to produce, How to produce whatever they do produce, and For Whom the good will be produced (the predictions). Without scarcity the answers would be trivial, or best dealt with in the context of the Garden of Eden. Almost all historical questions and all political problems require knowledge of how these questions are answered and/or how they should be answered. Distinguishing between the positive analysis of what is and the normative analysis of what ought to be should be, but seldom is, a central emphasis in all social studies as it is in economics. Students can then learn whether a controversy is one over facts, logic or values so that the "opinions" that so many questions "are all a matter of" are made explicit.

The two major reasons for confining the programs to a card programmable calculator such as the HP-65, 67 or 97, or the TI 58 or 59 are, first, characteristically high school social studies teachers are afraid of computers. However, if something can be put into your pocket and taken home at night to play with, like the card programmable calculator, the teacher becomes much more receptive to innovations involving, as does CAGE, the latest computational technology. Secondly, using calculators still restricts you to relatively short programs employing simple functions and ones which solve systems of simultaneous equations by approximations replicating what students do in thinking about economic problems. Hence, the programs themselves are much more easily understood by students and teachers who want to learn the mathematics and programming technique appropriate to economic analysis.

III. Illustrations From A Sixth Grade Experiment

Although it had not occured to me that CAGE would be useful in grade school teaching, Mrs. June Bing, a teacher of 6th graders at Rose Hamilton Elementary School, Centerville, Indiana asked me to try it out in her class during February and early March in 1978. Since neither of us could be sure that these simulation games would not be beyond the capacity of the students two precautions were taken: First, we planned tentatively only four classroom hours for the experiment. Second, I planned to combine my initial and final presentations with a slide and tape recording show taking the students on a hypothetical voyage to the isle of Gamekon somewhere, it appeared, in South America (which the students had been studying)--some conjectured that it was on Lake Titicaca--and, finally bringing them back. I knew that the students would enjoy that part of it but the actual economic simulations turned out to be so successful--and I enjoyed the children so much in my first teaching experience with students below 11th grade--that I finally devoted eight classroom hours to economics in addition to about three hours for the travelog and other activities (including a farewell party from the people of Gamekon when we finally had to return to the real world.)

Since there were 29 students in the class, it looked as if there might be enough participants to produce a competitive labor market. Once we had "arrived" on the island a map of it was shown to the colonists. Gamekon had precisely 29 farms on it. An unscalable mountain range separated the island into two districts: the Blue District on the east, and the Green District on the west. Running through the mountain range about in the center of the island is a narrow, easily defended valley where the headquarters of the Kids' Colonies Development Corporation (KCDC) was located with Mrs. Bing and I installed as local agents.

The Corporation had provided each colonist with a grubstake to tide him over to the first harvest, tools necessary to grow the all-purpose product, agrifacts, for which the island is renowned, and a tent--models were given the kids to put on their farms--which self-destructs after five years. The map showed prevailing winds from the NW. The best seed for the island's soil and climate grew where the land was at a high altitude and where it was dry. This information was given to the colonists who immediately began conjecturing secretively among themselves about which farms would be best to homestead.

Then the problem of how to allocate the farms arose. It was decided that since KCDC had been founded for the express purpose of letting kids set up the ideal societies their parents had never dreamed possible, the country should have a flag, a national anthem, a motto and a great seal reflecting these aspirations. The agents helpfully suggested that, since we were going to be a peace-loving nation, we should also have a predatory beast to represent us like all other peace-loving nations. So, contests were arranged with the award of merit points to winners ranging up to 25 for the best national anthem.

Unlike college and high school students playing the game, these 6th graders were serious in their idealism. Their motto was "May all the animals and people live in peace and joy as long as they live on Gamekon." The great seal reflected the same sentiment: "Republic de Gamekon," it read, "Peace-Trust-Love."

Winners were selected by public acclaim and given priority in choice of lots according to the number of merit points earned.

If anyone thinks that 6th graders are incapable of understanding the economic implications of the prevailing winds being from the NW with a mountain range running north and south, and the best seed being upland dry, let him contemplate this statistic: the Spearman's rank correlation coefficient of order of choice of lot and fertility of lot chosen was .93 with a standard normal variable of 4.82 indicating that there were 3-1/2 chances out of a million that the order of ranks was independent!

Deeds to farms were given out conferring ownership to the inhabitants selecting them. Colonists in the Blue District to the east had an official seal in blue on their deeds while those in the west had green seals on theirs.

Having allocated the lots the agents informed the colonists that it takes two cords of agrifacts per year to survive on Gamekon. Since the world price of agrifacts was $1000 per cord one would have to earn $2000 to survive. What, some ask, is to happen to those of us who maybe will find out their farms aren't that fertile? Just the question I was waiting for. The class discussed the issue and concluded that there would have to be "tests"--a controlled experiment--to find out how production on Gamekon changed when more workers were employed on a standard farm.

1. The Agrifact Experiment

The second class meeting started out with my developing the whole notion of conditional predictions and how, in economics, preferences, tastes, resources and technology are assumed to be given for the solution of the three basic economic problems of What? How? and For Whom? to produce. Economic theory is to show how the predictions change when the conditions do.

After considering briefly who makes these decisions in a market economy, we turned specifically to the question of what a businessman would have to know in order to decide how to produce whatever he thinks he might produce? After some exploration students came back with their answer of "know-how," or technology, and how much "the different things he used"--factors of production--"would cost." Clearly the first things we would have to find out about the island was its technology. We would have to run the experiments that had already been suggested in the previous class.

For the agrifact experiment students were divided into: 4 one-person teams, 2 two-person teams, 2 three-person teams and 2 four-person teams. (Two other teams failed to follow instructions and were disqualified.) Each team was given a ruler for each person and a supply of 9 by 12 inch sheets of construction paper each representing one strip of fallow land on a standard-sized farm of average fertility. To raise a crop of agrifacts the team had first to plow the field by drawing black lines horizontally one inch apart; second, harrow the land by drawing red lines vertically one inch apart, and, third, sow the the land by drawing green diagonal lines from lower left to upper right connecting all of the intersections of the black and red lines. Each intersection (up to a quarter of an inch miss!) was counted as a harvested agrifact. The object was to see which team could get the greatest output per person under the restriction that no person could have more than one colored pencil in his possession at any one time. This meant that after each operation was completed a one-man team would have to go to the desk and exchange the present pencil for the next; two-person teams could organize production on the basis of having two of the three necessary "tools" in their possession at all times while three-kid teams had all the tools required all of the time. Four-person teams, however, had more people than required to use the tools effectively.

When the twenty minutes allowed for the experiment were up it was obvious that the best twosomes were much more productive than the best loners, and it was claar that the best three-kid team was much more productive than the best four-person team. In fact, the latter actually produced less total output.

The students were eager to explain the outcome emphasizing that the one-person teams had to stop all the time to get other pencils when the three-person teams never had to. The best three-person team told how the girl who was best at measuring did only that while the two others drew the colored lines as fast as possible. The four-person teams seemed to get in each other's way all the time. It came to the students, therefore, as no surprise when they were told that economists claim that specialization and the division of labor increase the efficiency of production by saving the time of workers having to change over from one tool to another; by not having to provide duplicate sets of tools (capital) for each workers; by allowing people with different abilities to use their special talents; and, by breaking jobs down into simple repetitive tasks that allow workers to increase their dexterity in performing each. How Adam Smith saw that the latter aspect of specialization might result in producing a society of morons was easily comprehended by these 6th graders too.

When the data on the best teams' production related to size of team were run through standard statistical curve fitting programs on the calculator, much the best fit was a cubic equation* just as economists generally assume, (although I was hoping that a parabolic production function would be found because of certain simplifications of economic theory it permits. The next day I drew the "production function on the board. I showed that first, as more labor was applied to a fixed amount of land, there was a range of increasing returns followed by a range of decreasing returns and, finally, by one of absolutely decreasing returns this exposition seemed not to confuse the students so that I was emboldened to give them a precise formulation of the law of diminishing returns. How the students in the poorer teams could be expected to learn over time, if we repeated the experiment, to produce as efficiently at the best teams, with exact analogies in the real world, seemed not to strain their credulity either and introduced them to an important cause of economic growth.

2. Year 1--Feudalecon

After the discussion of the production function during our third class hour I distributed KCDC Form #10.0 "Production Functions in Gamekon" reproduced below. As you will see, there is no range of increasing returns--they are parabolic functions--because, I explained weakly to the kids, all economies were exhausted by the time one used a single kid-year of labor on a Gamekon farm. In return you will also see, just as most of the 6th graders did, that each additional worker produces two less extra cords of agrifacts i.e., output increases by a constantly decreasing amount--diminishing returns.

	Output in Cords/Yr. per Farm on the Grade of Land Shown				
Labor (Kid-Years)	α	β	γ	δ	ε
0	0	0	0	0	0
1	14	8	6	4	2
2	26	14	10	4	2
3	36	18	12	6	2
4	44	20	12	2	
5	50	20	12		
6	54				
7	56				
8	56				

A moment of silence signified the passing of Year 1. To find out what grade of land they had acquired the children were directed to look at an obscure corner of ther deeds. There they would find a number--14, 8, 6, 4 or 2--showing how much output they had produced for the first harvest. By looking at the production functions they then could also tell the grade of land they had acquired. The distribution of the farms was:

Grade	West	East	Total
α	2	4	6
β	2	2	4
γ	3	2	5
δ	3	2	5
ε	5	3	8
	15	13	28

One lot was up on the mountain side and inaccessable so that it could not produce any output. It was given to a student who was absent.

Keying into the calculator the data on the number of farms on each grade of land, and the price of agrifacts, the appropriate CAGE program produced the result that the Gross National Produce was $178,000. Another program showed that the lowest 1/5th of income earners (epsilons) received only 4.9% of the GNP--a situation worse than in the U.S.--while the highest 1/5th received 43%--even more than in the U.S. The Gini ratio--a measure of how much income distribution deviates from (.00) perfect equality--was .37, however, compared to the U.S.'s .39 (using the same linear approximation) because of less inequality in the middle income range on Gamekon. While I made these calculations in class with an astonished throng of 6th graders surrounding me, the teacher unfamiliar with economic theory would simulate the class's simulated economy using CAGE before ever stepping into the classroom.

---

* $y = 0.724 + 3.481 x^2 - 0.706x^3$ where y is agrifact output and x is standard labor years of input, plot size and fertility being constant. The coefficient of determination, measuring goodness of fit, is $r^2 = .999$

3. Years 2 and 3--Competecon

At the "All Colony Congress" convened at the end of the first year, and the beginning of the fourth class, the Game-kids assembled to consider what to do to improve their lot, especially that of the poor, starving epsilons. The meeting opened with the colonists facing the Gamekon flag and intoning the motto (a ritual that was to become a tradition in Years to come.) The deliberations showed that there was agreement that a labor market would have to be set up so that employers could negotiate wages with employees. So, the "Big Board" was established--a chalkboard at the rear of the classroom laid out as follows: WAGE at the head of the first column with amounts below running from $12,000 down to $2,000; the second column (about 3 feet wide) headed by LABOR DEMANDED with space beside each wage for prospective employers to put their names next to the wage they were willing to pay, followed by the number of workers demanded; and, last, a column headed by LABOR SUPPLIED for workers names at the wage they were willing to accept.

Before opening the market, I made several incantations over the calculator, which I called "The Wizard", keyed in the number of lots of the various land grades of the students present, the price of agrifacts and what I estimated would be the competitive wage for the economy. The Wizard was programmed to allow me to approximate by trial and error the wage at which the labor demanded would be exactly equal to that supplied, and how much labor would be demanded or supplied by each farm at that equilibrium wage. It also showed the maximum potential physical output and its value for the economy and how much would be produced on a farm on each grade of land. A second card took the total revenue per farm and subtracted total wages paid or received. The difference was business profit for those farms which either employed hired labor or on which owners were self-employed. These data were aggregated in the same program to predict maximum potential GNP, Total Wages and Total Business Profit. The predicted GNP was $302,900, and the predicted equilibrium annual wage was $6800.

My problem now was what should the Wizard's forecast be? A quick guess was: a maximum potential GNP of $270,000 ($9300 per capita) with wages averaging about $7000. I guessed that GNP would rise only by 52% instead of a possible 70% because of the lack of experience in making acute economic calculations on the part of my 6th graders. These forecasts were recorded and sealed in an envelope, then given for safekeeping to a teacher's assistant.

No instruction had been given the children on how to determine whether or not to hire labor or to hire themselves out to other employers. They had been told that a market economy (and the validity of the economic theory of markets) did not depend upon everyone knowing what to do but upon some people knowing what is in their interest and in others being able to learn from the former. And it was clear that many betas, gammas and deltas did not know whether to work for themselves or for others. Alphas and epsilons, on the other hand, were pretty sure what to do because the former were selected for motivation and intelligence by the merit point contest, and the latter were on the margin of starvation.

Labor markets in Gamekon sound like the Chicago Board of Trade on a day when disaster has struck a Brazilian coffee crop: both restore one's faith in cupidity as a motivating force for economic behavior. At the conclusion of the twenty-minute market session, the Wizard's forecast was read. That of an average wage of $7000 was not bad, but I overestimated the rise of GNP. Instead of the 52% I had guessed, the competitive economy experienced only a 39% increase to $248,000 as students with more productive farms hired students from the less productive ones.

I wish that space allowed me to tell you how a secret memorandum from the National Council for Blue Supremacy brought discrimination to Gamekon with "Blues", who owned most of the best land, refusing to hire "Greenies". I would also have liked to tell you how it came about from Year 2 on that it was the employers who not only became the richest, but also the poorest, islanders as they made good or bad business decisions and undertook risky ventures involving the renting of farms from other islanders.

Did the students learn any economics? On the Joint Council on Economic Education's Test of Elementary Economics their mean scores were 6% above the norming group's mean score on the 27 out of 40 questions we had covered in class. The norming group had, moreover, spent up to one half a year studying economics.

COMPUTER ASSISTED INSTRUCTION AND THE MENTALLY HANDICAPPED
SOME RECENT DEVELOPMENTS

H. J. Hallworth and Ann Brebner
The University of Calgary Faculty of Education
Computer Applications Unit
and
Vocational and Rehabilitation Research Institute

The objective of the project was to develop and assess the value of CAI in training the mentally handicapped. Potential advantages are that CAI facilitates continuity of individualised instruction using a carefully structured curriculum, and allows immediate reinforcement and remediation. Continua for social arithmetic and reading were developed following a task analysis of the skills relevant to the needs and interests of the mentally handicapped. These were implemented as two sets of CAI programs which are described together with their management systems. Special terminal equipment was designed to provide input and output facilities for mentally and physically handicapped persons. Results are given of controlled studies, and of experience gained over a period of six years. It is concluded that the project has been successful, with young adults and children, in developing higher levels of skill and increased attention span. Further, the costs justify the use of CAI for this population.

It is now accepted that the general objective of training the mentally handicapped should be to prepare them, insofar as practicable, for independent living in the community. Such an objective represents a complete break with the tradition of merely providing custodial care. Its basis, moreover, is both humanitarian and economic.

The humanitarian reason for training the mentally handicapped is easy to understand. The economic reason is also becoming obvious: persons supporting themselves instead of requiring expensive care throughout their lifetime save the community a substantial amount of money.

It has frequently been demonstrated that many mentally handicapped persons can be trained to support themselves and live independently (O'Connor and Tizard, 1956; Clarke and Clarke, 1974). The problem posed for the project described in this paper was, can computer assisted instruction (CAI) be developed as an integral and cost-saving component of such training? The answer, following six years of work, is in the affirmative. CAI is now being used successfully to train the mentally handicapped in Calgary. At the Vocational and Rehabilitation Research Institute (VRRI) it has been fully integrated into the curriculum and is effectively reducing costs.

Wherever training is provided for the mentally handicapped there is an urgent need for individualised instruction. A group of such persons typically exhibits a wide variety of learning histories, problems, present capabilities and rates of progress. This heterogeneity makes class instruction ineffective. A high ratio of teachers to students is therefore necessary and instruction is expensive. This situation raises the possibility that the retardate will become too dependent upon his teacher and never learn to make his own decisions. The type of instruction required places heavy demands on the teacher and there is frequently a high rate of staff turnover. This is particularly unfortunate since the mentally handicapped need a stable environment and continuity of instruction to ensure adequate progress.

In such circumstances, CAI is an appropriate technology. It provides a means of individualisation, gives continuity of instruction for each learner, and encourages the learner to become independent of his teacher. Further, it is much less costly than traditional forms of instruction.

A joint project was therefore undertaken by the VRRI and the Faculty of Education Computer Applications Unit at The University of Calgary. A time-shared computer dedicated to CAI is operated by the Unit. CAI terminals at the VRRI are linked to this machine by standard telephone lines. They now support a large and growing portion of the social education program provided for the 250 young adults attending the Institute.

Work has been extended into local schools for the mentally handicapped, in co-operation with the local Board of Education and a children's hospital.

Principles derived from the psychology of learning and instruction for the mentally handicapped have guided the project since its inception. Carefully devised training procedures have been developed which are based upon a well structured curriculum. Following a detailed task analysis, the material to be learned is presented in a series of small steps which are organised with a view to building up the required hierarchy of skills. The learner is kept active at the CAI terminal. Immediate reinforcement follows each response he makes and, where necessary, appropriate remediation is given after a wrong answer.

In the early stages of the presentation of new material irrelevant stimuli are removed and visual cues are used to facilitate learning. More stimuli are then introduced gradually, in order to assist generalisation by producing a situation comparable to that encountered by the learner in everyday life. Audio instruction and reinforcement have been found useful, but a minimal amount of audio output is provided because too many instructions merely confuse the mentally handicapped (Brown and Hughson, 1972). Considerable repetition is used to ensure overlearning, but care is taken not to give so much that it induces boredom.

It is important to motivate the mentally handicapped learner and, if possible, to increase his attention span. Experience shows that, with CAI, both motivation and attention span increase markedly. However, both are helped by reinforcement in the form of a star or tick drawn on the CRT, a picture shown by a slide projector, an audio message or the ringing of the bell on the CAI terminal. In contrast with the variety of messages provided for other learners, the reinforcement for the mentally handicapped follows a standard format which can readily be identified throughout the program.

Also, some of the programs make use of a cartoon personality with whom the users can readily identify.

Most important, however, is the fact that all learning materials are designed to have immediate relevance to the everyday life and needs of the mentally handicapped person. Such a person has frequently experienced failure in regular schooling and finds the usual school curriculum both frustrating and aversive. The curriculum is therefore designed to teach necessary living skills, and is divided into continua, each covering one set of such skills. A continuum is then organised as a series of units, each with criterion referenced measures. The sequence of units, and the learning materials within each unit, are developed in accordance with the principles described above.

At the present time two such continua are in regular use at the Institute. The first of these is concerned with social arithmetic and the second is in the area of reading. Although these continua follow the same principles they have considerable differences both in programming design and in the terminal equipment used.

The social arithmetic continuum is centred around money and introduces only those arithmetic skills needed for money handling in everyday living. This means that such topics as counting, addition, subtraction and place value are emphasised, while multiplication is only introduced towards the end, and division is not included.

The early units of the continuum cover counting skills, simple problems in addition, subtraction and place value, and the concepts of more and less. Next, the topic of money is introduced, first by units which teach the student to recognise each coin and attach a value to it. Groups of coins are then presented, and finally paper money. This section is followed by addition and subtraction involving three digits or more, and the concepts of "carries" and "borrows". The next units simulate the task of shopping to make a single item purchase. Finally, there are more units on buying; a series of units on banking, covering chequing and savings accounts; and a series on budgeting, both for food and larger items.

In total there are eighty units in the continuum, arranged in a strictly linear hierarchy. Each unit introduces the user to a particular skill and one hundred per cent mastery is required before he may proceed to the next unit. This ensures that the necessary overlearning is achieved, but at the same time creates the possibility of a student becoming trapped in a particular unit. To avoid this contingency, an over-ride option exists which enables the instructor to effect a transfer either forwards or backwards to any other unit. The linear sequence is then maintained from this new unit. The over-ride facility is also used when remediation is needed for previously learned skills.

The instructor monitors the users' progress through the materials by means of two types of report. The first of these lists the names and ID numbers of all users registered on the system, together with the name of the unit on which they are working and the date on which they were registered. The second is a detailed report for an individual student. It lists, for each unit attempted, the number of tries, whether passed, and the total time spend on the unit.

Within each unit, a non-didactic approach is used which attempts to shape and give practice in a new skill. However, the widely differing content of the units means that most require a unique program. These programs are called from a main management program which also controls operations on the student file structure.

The equipment used for these programs consists of a Teletype or cathode ray tube (CRT) terminal together with a computer controlled random access slide projector. The latter is required to display pictures of coins and bills, items to be purchased and different banking forms.

It is also possible to attach to this terminal configuration a special keyboard on which the keys are coins and bills. Use of this eliminates

144

some of the unnecessary mental operations which otherwise have to be performed before an answer can be entered. For example, in order to enter the answer of a "dime" on the standard keyboard, it is necessary to translate this to "ten", then to locate and press first the "one", then the "zero" and finally the "return" key. On the special keyboard only a single key is pressed, and this is a real dime.

The major part of the continuum on reading is developed in CAI form and the remainder is being programmed at the present time.

The continuum has three distinct parts. The initial programs teach recognition of letter shapes and, after a sequence of steps, recognition of identical words.

The main section of the sequence is a computerised version of the Clifton Audio-Visual Reading Materials (Brown and Bookbinder, 1967), an individualised reading program designed for the adolescent mentally handicapped population and used successfully in both England and Canada.

These materials consist of forty programs, each of which follows an identical format. The student is shown some letters or words, the corresponding sounds are given, and he is asked to speak these aloud. This is followed by a multiple choice test to determine which words he has learned.

As in the social arithmetic continuum, the hierarchy for presentation of the programs is linear with an over-ride option available to the instructor. Here, however, overlearning is achieved by repeating each lesson at three consecutive training sessions.

From a programming point of view this series is structured very differently from the social arithmetic programs. Only one program, accessing different data files, is required to present the units. At the present time the student files are extensive as each answer is recorded for research purposes. Consequently more detailed reports are generated.

The third section of the continuum is concerned with social sight vocabulary. These units are for those users who are not yet able to attempt formal reading. The students are taught to recognise and react appropriately to a limited set of words which they meet and need to understand in their everyday environment. This list includes such words as TELEPHONE, RESTAURANT, NO TRESPASSING and POISON.

In learning a social sight vocabulary even a small amount of audio greatly improves performance (Holz, 1976). In learning to read, the audio component is obviously of prime importance. Hence a multi-media terminal is used. This consists of an interface to which are attached a CRT terminal, a random access slide projector, an audio cassette player, and a touch sensitive display covering both the slide projector screen and the CRT. This display enables the user to respond to questions simply by pointing at the answer.

Further continua are being developed at the present time. These include one containing units on such topics as telling time and recognising laundry symbols, and others aimed at developing specific vocational skills.

Many of the mentally handicapped also have associated physical handicaps and for such persons specialised terminal equipment is in use. A high percentage of the cerebral palsied have poor vision and for this group output is displayed on a 23 inch monitor in order to enlarge the characters.

Special input devices include a POSSUM indicator board which consists of a 9 x 9 matrix of characters, any one of which can be selected by moving a light horizontally and then vertically on the board. Controls include a joystick, a dual rod control, a large hand switch, a foot switch, a head control and a pneumatic control. The device can be attached to any terminal configuration.

Another piece of special input equipment, which again can be used in conjunction with any terminal, is a POSSUM Expanded Keyboard. This has a full set of keys which are large, recessed, and widely spaced. There is also a delay mechanism which allows the user to press another key if he inadvertently presses the wrong one at the first attempt.

A new multi-media terminal based on a microprocessor is now being developed and brought into use. This has graphics capabilities which will allow the display of large characters, thus eliminating the need for the large 23 inch monitors. It is also envisaged that much of the reading continuum material at present on slides will be shown on the CRT, although some slides will still be required. The audio in this system is produced by means of a synthetic speech device and is expected to provide considerable versatility. It is anticipated that this will become the standard terminal used in the project to provide CAI for the mentally handicapped.

CAI is now established as an effective procedure for teaching mentally handicapped young adults and children at the VRRI and at special schools in Calgary. This has been amply demonstrated by controlled studies in the areas of buying (Strain, 1974), budgeting (Sandals, 1973), pre-reading (Eaton, 1975) and social sight vocabulary (Holz, 1976). In each case, however, the programs have been designed specifically to meet the needs of the mentally handicapped learner, and special terminal equipment has been used.

CAI is also comparatively inexpensive. The most recent calculation regarding instruction at the VRRI suggests that it costs approximately one-half as much as conventional training for the mentally handicapped, and the cost is still rapidly decreasing. CAI is therefore regarded as an acceptable tool whereby an instructor can supply a better quality of training to a larger number of persons. As more knowledge is gained regarding the design of CAI programs for the mentally handicapped, and as more advanced equipment is brought into use, the number of students per instructor is gradually being increased: the last such increase took place three months ago. The effect is to release more instructor time for those forms of training which require personal contact.

The results obtained to date represent no more than a beginning and plans are being made to extend the project. However, it is considered that a base has now been established from which more can be achieved in training the mentally handicapped by means of CAI.

REFERENCES

Brown, R.I. and Bookbinder, G.E. (1967). The Clifton Audio-Visual Reading Programme. Educational Supply Association: Harlow, U.K.

Brown, R.E. and Hughson, E.A. (1972) Verbal instruction in the performance of non-verbal tasks. Report to the Alberta Human Resources Research Council

Clarke, A.M. and Clarke, A.D.B. (1974) Mental Deficiency: the Changing Outlook. London: Methuen

Eaton, P. (1975) Visual discrimination and computer assisted learning. Unpublished Ph.D. thesis, University of Calgary

Holz, Else (1976) A study of the use of computer assisted instruction for teaching a social sight vocabulary to mentally handicapped adolescents. Unpublished M.Sc. thesis, University of Calgary

O'Connor, N. and Tizard, J. (1956) The Social Problem of Mental Deficiency. London: Pergamon

Sandals, L.H. (1973) Computer assisted learning with the developmentally handicapped. Unpublished Ph.D. thesis, University of Calgary

Strain, A.R. (1974) Computer assisted instruction in social arithmetic for the retarded. Unpublished M.Sc. thesis, University of Calgary

ACKNOWLEDGMENT

Part of the work reported in this paper is supported by a research grant from I.B.M. Canada Limited.

STUDENT ACCOMPLISHMENTS FROM INSTRUCTIONAL COMPUTING[1]

Beverly Hunter and Carol Hargan
Human Resources Research Organization

In what ways will our students benefit from instructional computing? We need answers to this question in order to: allocate funds for instructional computing; plan appropriate computing facilities for our schools; decide on priorities for applications; set up goals for instructional computing projects. This paper provides the following information: (1) Accomplishments of elementary and secondary school students as a consequence of instructional computing; (2) the kinds of evidence or documentation available from the schools and school districts with regard to these accomplishments; and (3) a framework to help encourage the sharing of such information.

PURPOSE OF THIS PAPER

In what ways will our students benefit from instructional computing? We need answers to this question in order to allocate funds for instructional computing; plan appropriate computing facilities for our schools; decide on priorities for applications; set up goals for instructional computing projects.

The literature on instructional computing is oriented towards a framework of computer applications: CAI, CMI, simulation, problem-solving, and so forth. This paper provides an alternative framework. We focus on students and their accomplishments as a consequence of using computers. This paper provides the following information:

1. Accomplishments of elementary and secondary school students as a consequence of instructional computing.

2. The kinds of evidence or documentation available from the schools and school districts with regard to these accomplishments.

Our purposes are to help educators and planners to set realistic goals for instructional computing, and to encourage documentation and sharing of information on student accomplishments.

SOURCES OF INFORMATION

Information on student accomplishments was obtained from thirteen public school districts and nine elementary and secondary schools that are participating in the project, "Exemplary Institutions in Academic Computing." These Districts and Schools are listed in Figure 1. These schools and districts were selected through the following process.

First, we conducted a systematic search for institutions that are regarded as outstanding in their uses of computers for learning and teaching. Invitations were mailed to seven thousand educators and technologists who belong to professional organizations concerned with educational computing. These individuals were invited to nominate one or more educational institutions that they regard as outstanding. Nominators were asked to give specific reasons why the school should be considered, given the objectives of our study.

Over 500 individuals responded, nominating 370 institutions that met our criteria.[2] Eligible institutions included individual elementary schools, secondary schools, school districts, community colleges, colleges, and universities, and public access institutions such as museums.

Second, we contacted, at each nominated institution, an individual who has a purview of instructional computing activities. In many cases, this individual is the Director of the Computing Center or a Coordinator of Instructional Computing. All but one of these individuals were happy to participate, and provided information about their activities via a telephone interview with a member of our staff. The product of this activity is an Academic Computing Directory,[3] published by HumRRO, that gives brief information on the reasons for nomination, the enrolment, typical computer applications, make and model of main computer(s), number of terminals, and contact information.

Third, the nominees were invited to respond to one or more of a series of open-ended questionnaires corresponding to the following Categories of Excellence.

1. Institutional Commitment to Instructional Computing
2. Student Accomplishments
3. Institution Productivity
4. Spectrum of Applications
5. Computer Literacy Programs
6. Computer Science and/or Data Processing Programs
7. Outreach
8. Model

These questionnaires required considerable work on the part of the respondents. By completing one or more of the questionnaires, an institution demonstrated its willingness and ability to share information. HumRRO staff and consultants then reviewed all candidate institutions within each Category of Excellence. We selected as Exemplars in each Category those institutions that provided complete answers and demonstrated a high commitment to instructional computing. The product of this third stage is a list of Exemplary Institutions distributed by HumRRO.

Each of the Exemplars in Category 1, Student Accomplishments, submitted information in response to the questionnaire shown as Figure 2. These institutions provided some evidence, whether statistical, descriptive, or both, of the successes of their students as a result of using computers. Our approach to the selection of exemplars was not rigid in the sense that we expected a formal research study. We looked at the information we received in terms of its usefulness to the institution. In other words, these were the data local decision-makers needed to continue support for computing activities.

NATURE OF THE DATA

The "evidence" provided by the twenty-two institutions was diverse, ranging from teacher opinion to formal evaluation studies. We found that in practice more decision-makers relied on descriptive information than on formal evaluations.

Eight school districts and one school submitted copies of evaluation reports. All but one of the reports described projects that provide drill and practice in reading and/or math for elementary and/or junior high students. Since the majority of these projects were Federally funded (Title I, III, or IV); the evaluation studies fulfilled a requirement of the sponsor.

The remaining schools and districts provided anecdotal information as well as sample student projects and documentation of various student activities, awards and employment opportunities.

CATEGORIES OF ACCOMPLISHMENT

The student accomplishments may be grouped into the following categories:

1. Improved scores on tests of basic skills
2. Increases in mastery of learning objectives
3. Acquisition of problem-solving skills
4. Improved attitudes towards school, learning, or self.
5. Projects that benefit school and community
6. Awards and scholarships
7. Employment

Examples of each of these kinds of accomplishments are presented in the following sections, along with information about the kinds of computer applications supporting the students.

IMPROVED SCORES ON TESTS OF BASIC SKILLS

Eight evaluation studies describe student achievement gains in reading, mathematics, and/or language arts. Features of these studies are summarized in Figure 3. There are several similarities among these studies--number and type of students, objectives, subject matter, funding, instructional software, hardware and results.

Students involved in these projects are mostly in grades 3-8 and were below grade level in their basic skills. Many are minority students or are from low income families. These characteristics of the students, coupled with the relatively large numbers of students involved, make the positive gains in test scores especially significant.

Instructional software in five of the eight cases was based on the strands approach originally developed at Stanford University. Montgomery County and Kansas City Public Schools developed their own CAI math programs. Woodridge Public Schools access the University of Akron's reading program.

Hardware involved in these projects included IBM 370, H-P 2000, and CCC A16.

The School District of Kansas City began academic computing in 1968-69 under an

ESAA Title III grant. Since 1973 the support for computing has come from local funds. All eighth graders in two junior highs have regularly scheduled CAI sessions. Thomas Hartley, CAI Coordinator, summarizes the program as follows:

"This area of basic math skills is one of particular weakness in the Kansas City School District. The median 1975 scores in grade equivalents for Kansas City eight graders on a fall administration of the ITBS was 6.4. This compares to 8.2 for national norms and 7.6 for large city norms. This program gives evidence of providing a significant boost to student achievement in this area. Therefore, the program has maintained a relatively high priority in the district and has survived several drastic forced budget reductions."

Over 1000 students in the Highline School System, Washington, receive CAI instruction every year. The objectives of their Title I project were:

"1. Students participating in mathematics classes supplemented by CAI will show one month's growth in computation skills for each month in the program as measured by a normative based standardized mathematics test and/or the Hewlett-Packard courseware for measuring student achievement of objectives.

2. Students participating in language arts classes supplemented by CAI will show one month's growth in language usage skills for each month in the program as measured by a standardized language arts test and/or the CCC procedures for determining grade level achievement and gains in terms of months.

3. Students who are deficient in reading skills and are receiving CAI reading lessons will show one month's growth in reading skills for each month in the program as measured by a standardized reading test and/or the CCC procedures for determining grade level achievement and gains in terms of months.

4. Students using the computer terminals for CAI will have positive attitudes toward this experience as measured on a questionnaire designed to measure attitude toward computer learning by averaging 75% of the maximum possible positive attitude score on this questionnaire. This objective was not included in the project proposal for the 1976-1977 school year."[4]

These student outcome objectives were met; data are presented in their Final Report. The personnel in Highline continue to produce a yearly evaluation study (now under no obligation to the government) because they believe the report is valuable for administrative planners and is informative and motivating for teachers.

Montgomery County Public Schools developed the OWN (Operation Whole Numbers) program and evaluated its effectiveness with children in grades 3-6. They found that the CAI students averaged from 1.1 to 3.6 months greater gain in mathematics achievement than the control students over fourteen months of school. They found this program to provide greater benefits for third and fourth graders,[5] rather than 5th and 6th.

INCREASES OF MASTERY OF LEARNING OBJECTIVES

Computer-managed instructional systems are being implemented in some schools, often to support diagnosis, prescription and testing in an individualized learning environment. Two exemplars submitted information on student accomplishments in a CMI environment. In both cases the proportion of students mastering the subject matter increased. At Jordan School District in Utah, GEMS[6] (Goal-Based Education Management System) monitors and guides student progress in science, math, reading and English. In 1977 the system was supporting approximately 7000 students and 150 teachers. Preliminary evaluation results in reading indicade that students using GEMS have significantly improved their achievement test scores

A CMI system was developed for freshman science at Alexis I. duPont High School.[7] The results of their evaluation of the system showed an increase in concept retention, and a marked rise in attitudes toward learning science for the students involved.

ACQUISITION OF PROBLEM-SOLVING SKILLS

Several of our Exemplar schools reported student use of computing in problem-solving activities in chemistry, physics, social studies, and mathematics. We would expect, therefore, to find information regarding the kinds of problem-solving skills the students acquire as a result of these applications. This was not the case. Only one of the schools described problem-solving skills as a student accomplishment, and this case involved students writing computer programs.

At Joyner Elementery School in North Carolina, the children learn to write programs in the BASIC language. Their teacher says the computer provides an environment in which the students learn to focus on a problem, to recognize and correct their errors, and to be precise in their solutions. These experiences are claimed to improve the students' study skills and their ability to solve problems in math, social studies, and language arts.

Problem-solving abilities of students are also reported indirectly, via student project descriptions.

IMPROVED ATTITUDES

Many schools and districts have done surveys on the attitude of students, teachers, parents, administrators, etc., toward instructional computing. The results of several such surveys are included in the reports prepared by Dallas[8], Memphis[9], Kansas City[10], and Highline[11].

Three school systems reported that students involved in computing showed improved attitudes toward learning, school and/or themselves. The students from all three districts are participants in CAI projects in reading (Los Nietos, Woodridge) and/or math (Los Nietos, Dallas). The students are all interacting with widely used CAI packages; University of Akron's reading package at Woodridge, CCC's math and reading at Los Nietos, and Hewlett-Packard's version of math strands in Dallas. The attitude-improvement data are sketchy, but they provide promising hypotheses for further research.

Both Dallas Independent School District and Los Nietos Elementary School District in California, report increased attendance of students involved in computing. At a middle school in Dallas the attendance is 14% higher among students in computing projects.

Vandalism is said to have decreased in the Los Nietos and Woodridge schools. Counselors in Woodridge report that the students coming from the remedial reading program have a "better self-image."

Dr. Nelson Crandell of Los Nietos believes that much of the achievement gains by the Los Nietos students can be attributed to the concept of "locus of control."

"This term refers to how the child perceives himself controlled. The internally controlled child perceives that he is in control of his own destiny, that his achievements are usually commensurate with his efforts, and that he can reach most goals that he sets for himself. The externally controlled child believes that his life is controlled by outside forces. He believes that what happens to him is the result of destiny, fate, luck, chance, the intervention of powerful others, or acts of God.

"It is quickly evident that the child who perceives that there is no relationship between his own efforts and his achievements is under a tremendous handicap. Many of these children believe that there is no way they can help themselves succeed so they spend their time in trying to avoid failure. They do this by avoiding themselves either physically or psychologically from most learning situations.

"The internal child, on the other hand, has learned to savor success. He spends his time putting himself in situations where he can succeed and once again know the pleasure that comes with success."[12]

Although Crandell believes that more research is necessary, he feels that interacting with the CAI programs have made the children become more internally than externally controlled. This, he believes, can account for increased reading scores by a child who is taking CAI arithmetic only.

PROJECTS

Students in computer science courses and computer clubs often undertake projects that benefit their own school or the community. Such projects were cited by fourteen of the twenty-two exemplars as evidence of their students' accomplishments. Projects included systems programs, instructional packages for other students to use, administrative programs and applications to solve local problems. The student programmers worked in many different environments with a variety of machines and software packages. This diversity contrasts with the similarities in hardware and software in the CAI projects.

At George Washington High School in Denver, Colorado, second year computer math students select a project of interest to them. These projects have produced programs for several departments at George Washington and for community organizations. Examples include:

- A carpooling program, cited as outstanding by the U.S. Department of Transportation, which aided over 500 corporations.
- Regression analyses to model sales tax revenues from 34 industries in Denver to help the City and County predict future revenues.
- Units for consumer math courses in the Denver Public Schools (e.g., savings plans, inflationary spirals).
- Programs for the English teacher to analyze Shakespearian plays, the psychology teacher to study biorhythms and the art teacher to examine the temperature for ceramic glazes.

The students use seven terminals in the math lab tied to Denver Public School's Univac 1130.

Belmont High School in Dayton, Ohio, offers an interdisciplinary social studies/computer science course that meets two hours/day, five days/week, for one year. The students learn about their local governments while using the computer to help analyze data, solve local problems, and compare forms of government.

At Hull High School in Massachusetts, the computer science students have analyzed various functions of the Fire and Police Departments. They have programmed a catalog of dwelling units for the Board of Health. They developed management inventory system for distributive education's retail store and prepared programs to generate statistics for all the high school teams.

At Riverdale County School, a K-12 private school in Bronx, New York, the students wrote their own version of the math strands (for the DEC PDP 11/45 or 11/34) developed at Stanford University, as well as the routines to manage and report on pupil progress through the curriculum. This package is being used by the elementary students at Riverdale and has been distributed to other schools. A Riverdale student presented this package at the 1977 National Computer Conference.

Students in Fairfax County, Virginia, under teacher supervision wrote a series of math CAI programs that are now used by approximately 4000 students each year.

The members of the computer club at both Syosset High School and Amsterdam Central Senior High School wrote instructional programs as a service to the school, including tailor-made CAI programs for teachers (e.g., to perform the tedious calculations of a chemistry lab, French and Latin drills, new library books, etc).

Ballou students in Washington, D.C. have written administrative programs for budget analysis, inventory, attendance, class rank, test scoring and analysis, and school banking.

The computer club at Sehome High School in Bellingham, Washington, enlisted resources within the community (as well as their own imagination) to improve their computing facilities and raise funds to purchase terminals and a microprocessor. The members, with the volunteer aid of the Air National Guard and the Army National Guard, were able to bury 2500 feet of donated cable to eliminate all telephone and modem costs. They have raised over $7,000 in the past two years from salvage, sale and trade of donated wire, and computer dances. In the Fall of 1977, a new course, Computer Design and Construction, was given at Sehome. Twenty-seven students and two teachers successfully built, debugged, and are now programming a Processor Technology SOL-20 microprocessor.

Students also produce programs to enhance the computer systems. At North Salem High School in Salem, Oregon, students programmed all the necessary revisions when a microprocessor was purchased. The advanced computer students have programmed many modifications to North Salem's DEC Classic and PDP 8I systems.

A students in the Richardson Independent School District in Texas, wrote a FORTRAN IV compiler which is used in the regional network, was purchased by Hewlett-Packard, and is now used all over the United States.

AWARDS AND SCHOLARSHIPS

One-fourth of the exemplars cited students whose computer projects were successful in science fairs and programming contests. For example, both individualsand teams from Richardson ISD have competed in contests with many of the winning students receiving scholarships. Science fairs now have categories that relate to computing. Four out of the seven finalists in the computer area of the Long Island Math Fair were students at Syosset High School.

EMPLOYMENT

Many schools and school systems have curricula in computer science and/or data processing. Data processing courses, in general, focus on job entry skills such as keypunch and computer operations, while computer science or computer math curricula emphasize programming and design concepts. Students from exemplar institutions have taken advantage of the skills acquired in such courses to find employment in computer-related occupations.

The Vocational Division of the Memphis City Schools implemented a data processing program in 1976-77. Almost 1000 students participated in one of the courses and of the 330 continuing to attain job entry skills, approximately 230 were placed in related positions.

Ballou High School is an inner city school in Washington, D.C. The student population is 99% black. Many of the students live in public housing projects and have parents on welfare. Ballou also houses a city-wide mini-school for science and mathematics. The school offers both data processing and computer science courses for over 300 students each year. Approximately 70 of these students per year obtain employment in one of the following jobs: keypunch operator, computer operator, tape librarian, computer aide, computer technician, computer programmer, programmer assistant, and office assistant.

In several institutions, students are the managers, operators and programmers for the school's computer facility. In Huntington Beach Union High School District present and former students provide the staff for instructional computing. The following is the hierarchy of student employment opportunities--each step conditional upon demonstrated ability and responsibility.

o Elementary and High School Cross-Age Tutor

o Exploratory Work Experience (School and District Computer Center)

*o Employed Work Experience (School and District Computer Center)

*o College Aide (Alumnae only) (School and District Computer Center)

*o EDP Intern (District Computer Center Only)

CONCLUSION

Data from twenty-two selected schools and school districts indicate a wide variety of ways in which students benefit from computing. The evidence ranges from a single teacher's observations to formal evaluation studies.

Eight of the twenty-two institutions provide formal evaluation studies showing improved scores on tests of basic skills in the elementary grades. Relatively large numbers of disadvantaged and minority children are gaining in their basic skills as a consequence of computer-based drills and tutorials.

Two of the institutions provided evidence that large numbers of students are increasing their mastery of the schools' learning objectives, with the support of Computer Managed Instruction systems that provide testing, diagnosis, and remediation.

Although several schools provide computer-based tools for students to aid them in problem-solving, little or no data were available to show what kinds of problem-solving skills the students were acquiring.

Improved attitudes towards school, learning, or self are reflected in attitude studies, attendance records, and decreased vandalism reports.

Student projects demonstrate contributions the students make to their school and community. These include student-written systems programs, instructional packages, administrative data processing programs, and computer applications to solve community problems.

Awards and scholarships for student projects in science fairs were cited by a fourth of the institutions.

Students gain employment both during school and after graduation as a consequence of acquiring skills in data processing. They gain experience as managers, operators, and programmers for their school's computer facility. More information on employment or college placement of graduates would be useful.

More documentation and dissemination of information about student benefits via computing, would be useful for a wide range of planning purposes. The framework provided here may help to encourage such sharing of information.

FOOTNOTES

1. This research was supported by a National Science Foundation Grant No. SED76-15399; Dr. Robert J. Seidel, Principal Investigator, and Beverly Hunter, Co-Principal Investigator. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2. Projects, consortia, thimesharing companies were not eligible.

3. Academic Computing Directory, First Edition. Alexandria, VA: Human Resources Research Organization, 1977.

4. "A Program to Meet Disadvantaged Students' Individual Needs for Basic Skill Development," Dr. Arthur L. Maser, Vern Johnson, and Jay Davis; Highline Public Schools Computer-Assisted Instruction Project Final Report (ESEA Title I), Seattle, Washington, June 30, 1977.

---

*Hourly stipend, others receive credit.

5. "Evaluation of Computer-Assisted Instruction 1975-76," by Catherine E. Morgan, Beverly J. Sangston, Robert Pokras. Rockville; MD: Montgomery County Public Schools, Division of Computer-Related Instructional Programs, March 1977.

6. "Summary TRACER Research Report, GEMS Project." Jordan School District, May 1977.

7. "A Computer Managed Approach to Individualized Instruction," by Gary E. Dunkleberger and Ruth W. Smith, Alexis I. duPont High School, October 1976.

8. "Evaluation of the CAI Mathematics Drill and Practice Program 1976-77," by James P. Papay. Dallas, TX: Dallas Independent School District, July 14, 1977.

9. "Final Evaluation Report for the Title I Computer Assisted Instruction Project," by Linda F. Harris. Memphis, TN: Division of Research, Memphis City Schools, June 1977.

10. "Summary of Evaluation Data on Computer Assisted Instruction in the Kansas City School District for 1975-76." Memo to Jack Casmer from Thomas A. Harley, Jr., June 16, 1976.

11. Op Cit.

12. "CAI: It's Role in the Education of Minorities." Whittier, CA: Los Nietos School District, September 28, 1977, (page 2).

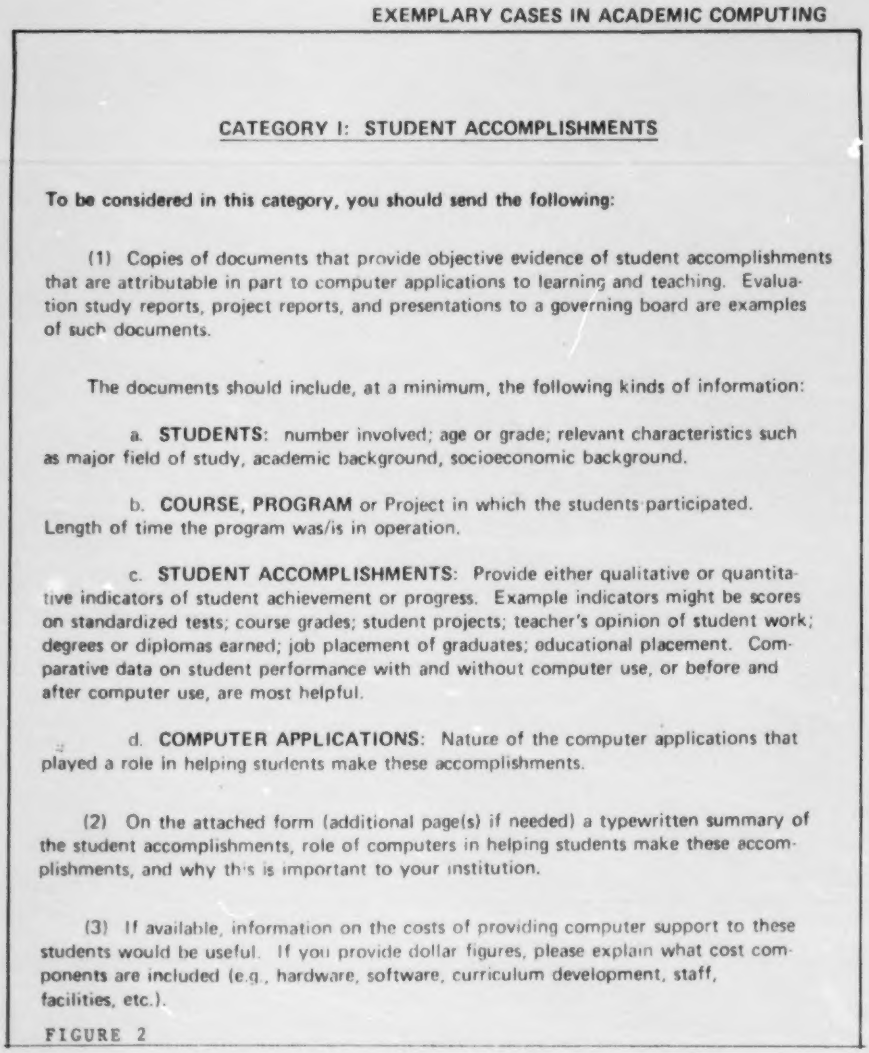
Elementary and Secondary Schools

George Washington HS (CO)
Ballou HS (DC)
Hull HS (MA)
Joyner Elementary School (NC)
Amherst Central Senior HS (NY)
Riverdale County School (NY)
Belmont HS (OH)
North Salem HS (OR)
Sehome HS (WA)

Public School Districts

Huntington Beach USD (CA)
Los Nietos ESD (CA)
Alexis I duPont SD (DE)
Montgomery County PS (MD)
School District of Kansas City (MO)
Syosset Central SD (NY)
Woodridge PS (OH)
Memphis City Schools (TN)
Dallas ISD (TX)
Richardson ISD (TX)
Jordan SD (UT)
Fairfax PS (VA)
Highline SD (WA)

FIGURE 1: EXEMPLAR INSTITUTIONS FOR STUDENT ACCOMPLISHMENTS



EXEMPLARY CASES IN ACADEMIC COMPUTING

CATEGORY I: STUDENT ACCOMPLISHMENTS

To be considered in this category, you should send the following:

(1) Copies of documents that provide objective evidence of student accomplishments that are attributable in part to computer applications to learning and teaching. Evaluation study reports, project reports, and presentations to a governing board are examples of such documents.

The documents should include, at a minimum, the following kinds of information:

a. **STUDENTS:** number involved; age or grade; relevant characteristics such as major field of study, academic background, socioeconomic background.

b. **COURSE, PROGRAM** or Project in which the students participated. Length of time the program was/is in operation.

c. **STUDENT ACCOMPLISHMENTS:** Provide either qualitative or quantitative indicators of student achievement or progress. Example indicators might be scores on standardized tests; course grades; student projects; teacher's opinion of student work; degrees or diplomas earned; job placement of graduates; educational placement. Comparative data on student performance with and without computer use, or before and after computer use, are most helpful.

d. **COMPUTER APPLICATIONS:** Nature of the computer applications that played a role in helping students make these accomplishments.

(2) On the attached form (additional page(s) if needed) a typewritten summary of the student accomplishments, role of computers in helping students make these accomplishments, and why this is important to your institution.

(3) If available, information on the costs of providing computer support to these students would be useful. If you provide dollar figures, please explain what cost components are included (e.g., hardware, software, curriculum development, staff, facilities, etc.).

FIGURE 2

School/ District	Population	Instructional Package	Hardware	Funding	Standardized Tests	Summary Results	Objectives
Highline SD	1200 students; Grades 5-9	H-P Math CCC Reading CCC Language Arts	H-P 2000	Title I	Comprehensive Tests of Basic Skills (CBTS)	Math Gains Elementary: 1.51 mo./mo. in program Secondary: 1.36 mo./mo. in program Languages Elementary: 2.22 mo./mo. in program Secondary: 1.10 mo./mo. in program	Positive attitudes toward CAI; 1 mo. gain/mo. in program
Los Nietos SD	1800 students; Grades 3-8	CCC Math CCC Reading CCC Language Arts	CCC A16	Title I	(CBTS)	Reading Gains 20-26 percentile points Math Gains 23-43 percentile points	
Kansas City	1000 students; Grade 8	Kansas City Math	IBM 370/135	Originally Title III	Differential Aptitude Test (DAT) SRA Achieve-Series	Gains of 9 - 19.9 percentile points 1.2 - 1.63 yr./yr. in program	
Memphis CS	1500 students; Grades 4-6	H-P Math	H-P 2000	Title I	Metropolitan Achievement Tests (MAT)	Gain of 1.3 - 2.4 mo./mo. in program	Mo. gain/mo. in program; Computational speed and accuracy
Dallas ISD	300 students; Grades 4-6 (one of several CAI projects)	H-P Math CCC Math	H-P 2000 CCC A-16	Federal funding for some projects	Shaw-Hiehle Computational Test	Gains for 6th graders; no significant difference for grades 4-5	
Woodridge PS	100 students; Grades 5-8 (remedial)	Univ. of Akron Reading	IBM 370/158	Title I	California Reading Achievement Tests	2 yrs. growth/yr. in program	
Montgomery County	2000 students; Grades 3-6	Montgomery County Math	IBM 370/158	Local	Specially Developed Tests	1.1-3.6 mos. greater gain than control students over 14 mos. (3.6 - 4.2 mos. for 3rd/4th graders)	
Belmont HS	15 EMR students; Grades 9-12	CCC Reading	H-P 2000	Local	Diagnostic Reading Skills, CBT McGraw-Hill	Increased comprehension and improved attitude toward reading	

Figure 3. **Summary of Evaluation Studies**

LANGUAGE LIMITATIONS FOR CAI

John F. Huntington
Miami University

CAI authors who must provide for frequently recurring instructional procedures become 'language bound' and subsequently tend to overlook equally important aspects of instructional programming. As a result, much CAI programming which employs general purpose languages remains at a relatively primitive level of instructional design. Additionally, programming requirements of general purpose languages practically necessitate program development by a competent computer programmer who is usually not also a competent instructional programmer.

This paper attempts to deal with the probable results which would be anticipated when general purpose and special CAI languages are used for instructional programming. It is essentially a comparison of the implications of choosing BASIC or Coursewriter.

Computer languages that can be used to program lessons for computer assisted instruction (CAI) can be included in a twofold category which is based on the relative powers of implicit algorithms of the language and the difficulty of learning how to apply the features of the language to lesson implementation. Two languages that exemplify polar positions of the dichotomy at the primary, prestructured level are BASIC and Coursewriter. BASIC is a general purpose, terminal oriented, conversational-like language which is easily learned while Coursewriter (IBM), which provides powerful functions related to instructional programming, requires intensive study to learn. The limitations which each language imposes on CAI program development are important considerations for systems managers. These limitations are:

1. The instructional resources required for staff to master the languages.
2. The power of the languages to provide for comprehensive, flexible and adaptive instructional programming.

General purpose languages like BASIC can be used to construct elaborate, sophisticated CAI programs but the instructional programmer must be highly skilled in programming lessons. There are few 'built-in' instructional functions in BASIC and coding for student record keeping must be continuously referenced in order to build a file of student responses and pathing. On the other hand, Coursewriter contains functions which can be called to perform record keeping in addition to implicit functions which construct student records. In Coursewriter there are many instructional algorithms which can be accessed by coding an operator with a modifier, while BASIC requires construction of subroutines by instructional programmers to perform the same tasks. From baseline development, BASIC requires considerably more planning and programming to reach the instructional programming level of Coursewriter with the advantage of being able to tailor the subroutines to local needs and with the disadvantage of requiring considerable staff development time to simulate a language which already exists. Though hardware considerations may dictate the choice of a language at present, it is probable that higher level languages like Coursewriter will be able to be used in the future on more systems as hardware capabilities evolve into more flexible and powerful configurations. When this state of development is reached, a real choice of languages to be used for CAI will exist. The choosing will involve primary considerations about whether to employ a language that features many 'build-in' instructional algorithms or to expend resources to design unique, local programs that may be more easily shared with others via a more universally used language. The following program statements using the Coursewriter II language developed for the late IBM 1500 instructional system represent a relatively high level of language specificity for instructional uses, CRT displays and specially designed hardware based on the

1131 computer. The numbers to the left represent segments and labels which would correspond to a program or subroutine coded in BASIC. The two character operating codes appear in the form of PR, DE, DT, etc. and work like operating statements used in BASIC. Modifiers follow operating codes and specify that certain special procedures be followed when executing the command specified by the operating code. For example, FN4 calls up a function that is named '4' and DTI calls for a display insert. Comparable modifiers in BASIC would be MAT INPUT or INV which cause a matrix to be created by input or to be assigned as an inverse of a square array. Note that modifiers for BASIC are not oriented toward instructional manipulations as are those for Coursewriter. The text or operands begin with either display position parameters or operations. A DT (display text) is followed by printing position parameters, then delimiters and finally by the actual text. In the case of an AD C2¬/C3, counter C2 is added to counter C3. Display parameters are automatically provided in BASIC with the result that programming is easier but flexibility is diminished. The letters and numbers to the right are statement numbers like those in BASIC which appear at the left of each program statement.

Coursewriter was designed to implement answer analysis, performance recording, timing control, implicit and explicit branching, buffers, counters, switches, labels, return registers, restart points, macros, and functions. The performance recording facility may provide:

1. The pathing taken by each student.
2. The student's name and the name of the course.
3. The time required to complete the response.
4. The actual response accompanied by a response identifier.
5. The type of response (correct, incorrect, etc.).
6. The contents of counters, switches, etc.[1]

It should be evident that provisions for these procedures must be programmed when BASIC is used since they are not part of the support software usually provided when BASIC is available on most systems. The issue that arises raises the question

AN EXAMPLE OF COURSEWRITER CODING [2]

```
1C2                                                               JH010010
P11A                                                              JH010020
      PS                                                          JH010030
      DE   0¬/32                                                  JH010040
      DT   10,11¬/¬/¬/<INTFLLIGCNCF TEST>                         JH010050
      DT   13,16¬/¬/¬/<PROGRAM><4>                  <JFH> [redacted]  JH010060
      DT   24,1¬/¬/¬/<B><E>XPFRIMENTAL                            JH010070
      PA   30                                                     JH010080
      DE   0¬/32                                                  JH010090
      DT   6,6¬/¬/¬/<T>HE USE OF THIS PROGRAM IS LIMITED          JH010100
      DT   9,2¬/¬/¬/TO AUTHORIZED INDIVIDUALS. <T>YPE THE         JH010110
      DT   12,2¬/¬/¬/CODE WORD TO CONTINUE.                       JH010120
      EPI  16,6¬/2,16¬/3,6¬/100¬/¬/                               JH010130
      NX                                                          JH010140
      DE   0¬/32                                                  JH010150
      DT   6,5¬/¬/¬/<Y>OU ARE NOT AUTHORIZED TO CONTINUE.         JH010160
      PA   30                                                     JH010170
      DT   10,2¬/¬/¬/<PLEASE SIGN OFF>                            JH010180
      PA   30                                                     JH010190
      DE   0¬/32                                                  JH010200
      BR   PRT1A                                                  JH010210
      DE   0¬/32                                                  JH010220
      CA   SIT                                                    JH010230
      DE   0¬/32                                                  JH010240
      DT   2,6¬/¬/¬/<R>EAD THE FOLLOWING DIRECTIONS               JH010241
      DT   5,4¬/¬/¬/CAREFULLY.  <PRESS THE SPACE BAR WHEN         JH010250
      DT   8,4¬/¬/¬/YOU WISH TO CONTINUE.                         JH010260
      EPI  8,Y¬/¬/1,39¬/9999¬/1¬/DUMMY                            JH010290
      DE   0¬/32                                                  JH010300
      DT   6,6¬/¬/¬/<T>YPE YOUR NAME.                             JH010310
      EP   10,2¬/2,10¬/35,2¬/¬/¬/10                               JH010320
      LD   80¬/81                                                 JH010340
      DE   0¬/32                                                  JH010350
      DT   6,6¬/¬/¬/<T>YPE YOUR AGE (IN YEARS ONLY).              JH010360
      EP   10,2¬/2,10¬/2,2¬/¬/¬/20                                JH010361
      FN4  EI¬/¬/¬/C2¬/S2                                         JH010362
      MP   12¬/C2                                                 JH010390
      DT   13,6¬/¬/¬/<T>YPE THE NUMBER OF MONTHS SINCE            JH010400
      DT   16,2¬/¬/¬/YOUR LAST BIRTHDAY. <I>F 15 OR MORE DAYS     JH010410
      DT   19,2¬/¬/¬/ARE A REMAINDER, ADD ONE MONTH.              JH010420
      EP   22,2¬/2,22¬/2,2¬/¬/¬/21                                JH010421
      FN4  EI¬/¬/¬/C3¬/S3                                         JH010422
      AD   C2¬/C3                                                 JH010480
      PA   10                                                     JH010500
      DE   0¬/32                                                  JH010510
      DT   2,5¬/¬/¬/<W>HAT NAME DO YOUR FRIENDS CALL YOU?         JH010520
      EP   10,6¬/2,10¬/20,6¬/¬/20¬/0                              JH010530
```

AN EXAMPLE OF BASIC CODING[3]

```
0001 REMARK THE FOLLOWING PROGRAM IS DESIGNED FOR THE CHINESE SPEAKING
0002 REMARK STUDENTS WHO STUDY ENGLISH AS A SECOND LANGUAGE. THIS
0003 REMARK PROGRAM CONCENTRATES UPON THE SIMPLE PRESENT AND SIMPLE
0004 REMARK PAST FORMS OF VARIOUS IMPORTANT VERBS,AS WELL AS UPON THE
0005 REMARK THE AGREEMENT BETWEEN THE VERB AND THE SUBJECT. THUS THE
0006 REMARK STUDENTS CAN ACQUIRE THE ABILITY OF ASSOCIATING THE CORRECT
0007 REMARK FORM OF THE VERB WITH A PARTICULAR SUBJECT AUTOMATICALLY.
0008 PRINT "HELLO, I'M A COMPUTER. WHAT'S YOUR NAME?"
0009 INPUT N$
0010 PRINT "NICE TO MEET YOU, ";N$
0011 PRINT "I'M GOING TO HELP YOU REVIEW THE SIMPLE PRESENT AND SIMPLE"
0012 PRINT "PAST FORMS OF THE VARIOUS IMPORTANT VERBS."
0014 PRINT
0015 PRINT "THIS PROGRAM IS DIVIDED INTO TWO PARTS:"
0016 PRINT "     PART 1: THE SIMPLE PRESENT TENSE  "
0017 PRINT "     PART 2: THE SIMPLE PAST TENSE     "
0019 PRINT
0020 PRINT "EACH PART CONTAINS TEN QUESTIONS."
0022 PRINT "ANSWER EACH QUESTION JUST BY TYPING IN THE CHOSEN NUMBER."
0024 PRINT
0025 PRINT "REMEMBER TO SAVE THE RUN OF THIS PROGRAM AND BRING IT"
0026 PRINT "TO YOUR TEACHER SO THAT HE OR SHE WILL KNOW HOW YOU'RE"
0027 PRINT "DOING AND HELP YOU WITH YOUR PROBLEM."
0032 PRINT "NOW, LET'S GET STARTED."
0034 PRINT "WHICH PART ARE YOU INTERESTED IN?"
0036 INPUT P$
0037 IF P$="PART 1" THEN 0039
0038 IF P$="PART 2" THEN 0773
0039 PRINT
0040 PRINT
0044 PRINT "1. HE ----- HIS WORK."
0045 PRINT "      1 TO DO "
0046 PRINT "      2 DOING "
0047 PRINT "      3 DO    "
0048 PRINT "      4 DOES  "
0049 PRINT "      5 DONE  "
0055 INPUT A1
0056 LET I1=1
0057 PRINT
0058 IF A1=4 GO TO 0181
0060 LET W1=W1+1
0061 IF A1=1 THEN 0069
0065 IF A1=2 THEN 0087
0067 IF A1=3 THEN 0098
0068 IF A1=5 THEN 0113
0069 GOSUB 0071
0070 GO TO 0120
0071 REMARK SUBROUTINE 'INFINITIVE' STARTS HERE
0073 PRINT "THINK CAREFULLY, ";N$;" 'TO + PRINCIPAL VERB' IS AN"
0074 PRINT "INFINITIVE. EVERY SENTENCE NEEDS A SUBJECT AND A VERB. SO.."
0075 LET S1=S1+1
0076 RETURN
0087 GOSUB 0089
0088 GO TO 0120
0089 REMARK SUBROUTINE 'PRESENT PARTICIPLE' STARTS HERE
0090 PRINT "NO, THE WORD YOU HAVE CHOSEN IS THE PRESENT PARTICIPLE."
0093 PRINT "THE PRESENT PARTICIPLE FORM CANNOT BE USED BY ITSELF"
```

of the importance of purchasing implicit instructional functions as part of a CAI language or developing them as subroutines, chained programs, etc. by using local instructional resources.

The experiences at some universities have led directors of CAI development activities to establish elaborate schemes to encourage content specialists to use CAI languages. When a language like Coursewriter is used, directors found that university professors were reluctant to learn a rather complicated language while users of BASIC faced professors who perceived instructional programming as little more than drill and practice since that was all directors could demonstrate in the way of utilization of the language for instruction. The response from directors using Coursewriter was to hire instructional programmers who were generally retreaded general programmers or newly trained staff who didn't know much about instruction or how to set up for programming instruction. Added to this problem was the discovery of the reality that most professors didn't know much about programming instruction either. The subsequent products from these teams revealed inadequate applications of effective instructional programming. The gap between computer programming and subject content was never satisfactorily bridged to enable instructional nethodologies of more than primitive levels to be applied to CAI lessons. One university which used CAI for medical school training found the problem so staggering that a system of several I/O strategies involving color coded forms representing true-false, multiple choice, completion, etc. were circulated among faculty so they could construct item stems, alternatives, distractors and scoring keys which were later programmed into logical sections by programmers. The interfacing of this procedure into a comprehensive course design is comparable to the analogy of the left hand being largely unaware of what the right hand was doing. The problem contered about a breakdown between the instructional programmer and content specialist. In order not to impose too much required learning of instructional logistics and methodologies on professors and to expedite programming that previously was bogged down by explanations tendered by programmers to professors and from professors to programmers, the solution of narrowing the channels of communication helped to get programming accomplished but deterred from the probability of obtaining richer instructional procedures. The choice of Coursewriter and the decision not to employ the bridging specialists that were knowledgable about programming instruction led to a systems breakdown and a response that diminished the potential of CAI. Subsequently CAI was used at this university as a back up, voluntary enriching activity rather than a primary form of instruction as originally planned.

Another university using Coursewriter experienced great difficulty attracting faculty to use the CAI system. Several instructional programmers were employed to code in Coursewriter after consultation with professors from various departments. A very similar problem of working out the relationships and roles of professors and programmers arose. To date, the most effective and sophisticated CAI lessons constructed at this university have been designed, coded, edited and maintained by individuals rather than teams composed of members with varying responsibilities. These experiences point out the advantage of learning Coursewriter and having knowledge of programming instruction and also being a content specialist. Until efficient teaming arrangements can be worked out this appears to be the best management arrangement when using a language like Coursewriter. The problems generated by lack of continuity due to staff turnover, the intimidation of content specialists by the requirements for learning the language, the inadequate attention to programming instruction and the difficulties associated with coordination and information exchanges between team members necessitates very careful systems planning to enable an institution to derive more of the significant potentialities offered by a language like Coursewriter. Failure or inadequate development of CAI programs is not due to a deficiency of the language but is attributable to instructional systems deficiencies.

While the potential for coding elaborate lessons is near at hand when Coursewriter is available, the choice of BASIC is one which also carries a commitment to design functions at the local site. These functions are provided by IBM as control facilities in Coursewriter. An example of a useful control function in Coursewriter is FN2 KEYL (function 2, keyletter) which permits answer analysis on the basis of matching selected characters of a CA (correct answer). This provides for treatment of spelling errors by using a maximum of two operating codes. In BASIC, a subroutine or several matching algorithms would have to be coded locally to perform the same function. The important point is that Coursewriter provides special functions as well as major and minor aspects of operating code sequencing that have been built with special references to instructional procedures. The implicit functions that result in the generation of student records would have to be simulated as special subroutines or chained programs using files in BASIC. Unfortunately, on many BASIC systems files cannot be used adequately for building cummulative records of student performance when several sign on/ sign off sessions are required. Too much erasing of existing contents of files occurs for many BASIC systems, especially those maintained on relatively small computers.

CAI specialists using BASIC must take a different approach to course development. It appears that the initial investment of the instructional resource should be to design and code subroutines and file management procedures that would permit individuals interested in CAI applications to benefit form a collective investment in systems potential. Essentially, this is the procedure used by IBM to provide these features for Coursewriter. The advantage of local development is increased tailoring of the system to user needs but at significant cost. It is not difficult to announce the availability of a BASIC system for CAI purposes but it is difficult to obtain adequate course development without team support. The history of the use of BASIC for CAI has indicated that the quality of lesson development is more dependent on the skills and persistence of individuals than on the provisions of institutions. Eventually this state of affairs must evolve into an institutionalized CAI development system to provide continuity, individuals skilled in coding and programming instruction and efforts to construct additional specialized language functions in order to permit content specialists to focus on their areas of competence. The alternative is to encourage teachers and professors to become facile in computer programming, systems management, programming instruction as well as knowledge of content, a defaulting procedure that has occurred at most institutions and that has resulted in stacking the odds against cummulative CAI development.

There are several levels of courseware methodology that range from primary to transitional to advanced.[4] a comparison of levels and models is presented below. An examination of the levels of courseware methodology should lead to the conclusion that Coursewriter and BASIC interface with methodologies at the primary level and the prestructured model. The important difference is that since BASIC is a general purpose language, it can be used by skilled programmers with adequate hardware configurations to construct generative lessons.

LEVELS OF COURSEWARE METHODOLOGY

Level	Model	Features
Primary	Prestructured	Content, objectives, strategies, etc. are prestored
Transitional I	Generative	Elements and/or rules are stored and used to build strategies after initial unput
Transitional II	Adaptive	A generative system with strategies under means-ends rules that permit evaluation.
Advanced	Self Improving	An adaptive system which can alter strategies when a student is not responding as expected.

Coursewriter conceivably could be moved to this level by writing special macros. In either case, the level of design and programming is high and the investment of staff time is extensive. Coursewriter although categorized at the primary level is much closer to the transitional I level than BASIC. It should be evident that CAI languages as they currently and generally exist are really not very sophisticated in providing for simulations of human teaching. Some of the work being done with generative CAI[5] and interpretations of natural languages by computers[6] should provide the lead for CAI systems to move from prestructured models to more elegant, higher level states. The choice of a language for CAI use is important in determining whether a local center can move toward more elaborate and successful uses of computers for instruction.

REFERENCES

IBM 1500 Coursewriter II Author's Guide. IBM Corporation, San Jose, CA.

Huntington, J.F. INTEL. A demonstration program for Coursewriter II. Miami University, Oxford, OH. 1978.

Huntington, J.F. HSU1. A course project for CAI. Miami University, Oxford, OH. 1978.

Huntington, J.F. CAI Courseware and Levels of Cognitive Development. Midwest Conference of Teachers of Educational Psychology, KY. 1976.

Uttal, W.R., et al. Generative CAI in Analytic Geometry. Newburyport, Mass: Entelek, Inc., 1970.

Winograd, T. Procedures as a Representation for Data in a Computer Program for Understanding Natural Languages. Ph.D. Thesis, MIT, 1970.

TICCIT vs LECTURE: AN EVALUATION OF TICCIT COMPUTER-ASSISTED ENGLISH COMPOSITION AND MATHEMATICS INSTRUCTION

Michael C. Jones
Hazeltine Corporation

In its first two years of operation, the TICCIT™ CAI system proved to be extremely successful. A major independent evaluation conducted by ETS and subsequent evaluations by TICCIT users have demonstrated that college students who completed "mainline" TICCIT math and English composition courses scored significantly higher on posttests, including essay posttests in English courses, than did students completing lecture sections of similar courses. The evaluations also show the importance of the instructor's role in managing student progress in an otherwise individually-paced TICCIT environment.

BACKGROUND

In order to investigate the impact of CAI in a community college setting, the office of Technological Innovations in Education at NSF awarded a multi-million dollar contract to the MITRE Corporation to develop and demonstrate TICCIT. The MITRE Corporation then engaged instructional design and CAI experts from the University of Texas CAI laboratory and Brigham Young University (BYU) to work closely with its own system engineers in the design, development, and implementation of the TICCIT CAI system. MITRE engineers were primarily responsible for the hardware and software development; the Institute for Computer Uses in Education at BYU, which then included those who were the key members of the University of Texas development team, directed the development of the freshman-level mathematics and English courses used in the demonstration. Assisting in the specification of those course materials were English and math faculty from both of the community colleges selected as the demonstration sites, Phoenix (Arizona) College of the Maricopa County Community College District and the Alexandria Campus of the Northern Virginia Community College.

SYSTEM DESCRIPTION

An important aspect of the project was the effort expended developing TICCIT's reliable and inexpensive display system. A brief physical description of the system must include the fact that the system is powered by a pair of minicomputers which direct the retrieval of various types of information, including digital text/graphic displays and audio messages, to be broadcast to the student's terminal, which includes a standard Sony television set, student keyboard, and optional light pen. The use of the television permits an extensive use of color in the digital displays as well as the use of full-color motion video (e.g. video tapes) and broadcast television.

The TICCIT project was concerned not with the development of a new hardware system but of a new instructional system. Therefore, a significant portion of the TICCIT project research was directed towards the development of an effective individualized instructional delivery model. The resulting model is a unique combination of new ideas and highly tested instructional design principles, some of which are briefly described below.

Mainline courseware. TICCIT is not designed to be used as an adjunctive resource to supplement regular classroom instruction but to be used as the primary source of the delivery of instruction.

The role of the teacher is intended to be that of manager-advisor, working with individual students in tasks such as selecting goals, diagnosing problems, prescribing special assignments, holding limited classes in subject areas causing particular difficulties, and, in general, providing the motivation required to overcome inherent pacing problems in such an individualized system. TICCIT administers and evaluates all primary instruction and lesson tests, provides students with inter- and intra-lesson advice concerning learning strategy, and provides instructors with student performance data.

Mastery. The testing on TICCIT requires all students to attain a specified level of achievement in each lesson. Each student spends as little or as much time as necessary to reach that goal; one aim of the system is to provide for each objective enough material at differing difficulty levels to allow students of divergent entry levels to achieve mastery.

Separation of Content and Strategy. The analysis of research conducted in connection with TICCIT and other projects indicated that the effectiveness of a particular learning strategy is independent of subject matter. The model of instruction implemented on TICCIT and found to be most effective in promoting mastery of the types of objectives used in academic course work is based, among other things, on the presentation for each objective of a generality statement, examples of the generality applied to specific instances, and practice problems. The example and practice items are present in large numbers allowing the student to choose from a divergent set of instances at differing difficulty levels. In addition, every generality and instance is further explained in detail through the use of "feedbacks" and "helps."

Not only does the separation of content and strategy provide the student with a consistent learning environment for each objective, but it also eliminates the need for the subject matter expert/author to be involved in lengthy determinations of teaching strategy. The built-in learning strategy makes possible a team approach to authoring, allowing a differentiated staffing scheme for cost-effective courseware production. Those highly trained in subject matter and instructional design perform only creative tasks, such as lesson specifications; routine formatting and data entry are completed by a specialized production staff.

Learner Control. Within the structured design of instruction, the student exercises control over the pace of instruction, the sequence of instruction, and the strategy of the instruction for each objective. From all of the instructional components available, the student chooses the type and quantity he/she wishes to see. The student is assisted in this responsibility by an on-line "advisor" which attempts to inform the student what options would appear to be reasonable based on the student's past performance.

EVALUATIONS--INTRODUCTION

Although original NSF project plans called for evaluations of costs, technical achievements, and educational impact, the final ETS report is concerned only with the latter. In order to judge the system's impact upon education at the community colleges ETS compared students in traditional classes with students in the same courses taught on TICCIT. Because of the mainline nature of the TICCIT courses, the comparison was sharply defined; students on TICCIT, although still directed by an instructor, generally received all or nearly all of their instruction on-line, while non-TICCIT students were instructed in the traditional manner, receiving instruction from textbook and lecture. Although the evaluators had no control over class composition, students often chose TICCIT or non-TICCIT sections on what might be considered a semi-random basis, and in the evaluator's estimate, class composition did not appear to create any bias toward either the TICCIT or non-TICCIT condition. The measures of educational impact included student achievement, completion rates, student and faculty attitudes, and student and faculty activities. In this paper, only the measures of student achievement and completion rates are discussed.

Although the ETS report does not contain data concerning TICCIT usage at Brigham Young University, many students were also enrolled in TICCIT math and English classes at BYU during the demonstration period. BYU conducted its own evaluation of the effectiveness of the system (Bunderson, 1977). The study at BYU was certainly not on the same scale as the ETS study, but the data from the BYU study has been extremely useful because of the several differences between the BYU implementation of TICCIT and TICCIT implementations at demonstration community colleges.

STUDENT ACHIEVEMENT

Measures. In order to compare the effects of TICCIT upon cumulative learning, ETS administered identical posttests to all students finishing the identified math and English courses; differences in test scores between TICCIT and non-TICCIT sections indicated the effect of the TICCIT program upon achievement, with an effect of 4 points representing an approximate difference in outcomes of ten percent of the test's total possible score. For math courses, the evaluation looked not only at the total score across items which tested objectives common to both treatment conditions, but it also examined achievement in terms of subscores for ability levels, such as problem solving ability, and content categories, such as linear equations. For English courses the posttest was divided into two distinct parts, an objective test of writing skills and an essay test.

Because of the apparent different natures of cumulative and immediate learning, the ETS evaluation included provisions for testing immediate learning in the math course. The measure for such learning consisted of a topical test to be administered to the students as soon as the material to be tested was completed by the students.

At BYU, posttests were not specifically designed to compare TICCIT and lecture as was done in the ETS evaluation. Instead, departmental exams which had been designed for the traditional classes in English and math were also given to the TICCIT students, although the curriculum designs were not identical. The BYU study also compared the effectiveness of the TICCIT English course on the basis of student performance on a university-administered General Education English Composition Exam.

Populations. In reporting the ETS achievement data for English sections, comparisons are not made between the entire group of students enrolled in the two conditions. In both TICCIT and lecture sections, less than the entire group of enrollees finished the course, and neither of the groups that did finish was representative of the entire population enrolled in the respective course. For both conditions, the entrance ability of those finishing was greater than those not finishing. However, compared to each other the groups finishing courses were similar; therefore, student achievement comparisons were between the students completing English on TICCIT and those completing lecture sections.

In mathematics at the community colleges fewer students completed the course on TICCIT than did in lecture sections and the average entrance ability of the students finishing within one academic term was higher than that of the students finishing lecture sections. Therefore, the initial comparison for math posttest achievement results was between the group completing TICCIT within one term and similar students (based on entrance ability) completing lecture classes. For additional posttest analysis, the population was substantially enlarged by including the scores of the many students completing TICCIT courses during a second academic term. A larger comparison group was also used in the analysis of topical test achievement.

Results--Math. A number of different types of analyses of the math achievement are contained in the ETS report. The first analysis involved eleven comparisons of TICCIT math sections and lecture sections of the same course taught during the same academic term. As described in the previous subsection, the posttest scores of students beginning and completing a TICCIT course within one term were compared to the posttest scores of lecture section students of the same entrance ability, where entrance ability is measured only by the pretest administered at the beginning of the term. In eight of the eleven cases, the TICCIT students scored higher on the posttest. Levels of statistical significance were seen in only two of the eleven cases; both cases of significantly higher scores favored the TICCIT students. Such results are encouraging but certainly not overwhelming.

However, it must be recalled that the TICCIT courses were "self-paced" and, that no external schedules to complete within one term were imposed upon the TICCIT students. As in many other self-paced environments, completions within a single academic term were low in number among the TICCIT sections. To expand the number of student observations being considered, the second ETS analysis dropped the time limitations of a single academic term. There were six different math courses at the two community colleges that were so analyzed. In other words, the achievement of all students who finished a given TICCIT math course during the academic year was compared to the achievement of similar students finishing a lecture section of the same course. In that analysis there were significant achievement differences in five of the six courses. All five favored TICCIT.

(Table 1 contains a small sample of the data from the first analysis, achievement within a single term. The table lists pretest and posttest scores for several terms of two of the observed math courses. Given are the pretest scores of all students who registered and took the pretest, the posttest scores

164

of all students who took the posttest, and finally, a pairing of the scores of students who took both the pretest and posttest.)

A third analysis in the ETS report examined ability and content subscores for math posttest to discover particular strengths or weaknesses in TICCIT instruction. Table 2 lists the ability levels and content categories analyzed on the basis of various posttest items. As might be expected, the number of items for each type of ability or category is small. ETS analysis indicated that although some of the subscores did approach reliability for group measures, most did not, making any outcomes more descriptive than conclusive. Few of the categories resulted in significant effects; only two significant effects were repeated across terms, and both were in favor of TICCIT. Solution of routine problems as an ability level involved positive TICCIT effects in every case and significant effects in two courses. The other repeated significant positive effect involved the content category of equations. These specific strengths must be attributed to the depth of coverage of each topic in a TICCIT course and the extent of practice opportunities. In general, in those areas of the math curriculum considered to require the highest ability and to be most advanced, TICCIT outcomes showed consistently positive effects.

The topical tests administered and the analysis of the effect of TICCIT are shown on Table 3. Generally, the effects follow the pattern found in posttests. Although only one of the results was significant at acceptable levels, in four of the cases the results show a sizeable point difference which represents a positive TICCIT effect equivalent to ten percent of the total possible score on the tests. Another indication of the positive impact of TICCIT upon achievement is the fact that these results apply to a wider range of students than those taking a posttest since the topical tests were administered earlier in the term. (The important value on the table is the t-statistic. ETS found significance when the absolute value of the t-statistic exceeded 2.0. A positive value favors TICCIT; a negative value favors the lecture condition.)

Results--English. Only cumulative learning as indicated through posttests was analyzed by ETS for achievement effects in English courses. The posttests themselves were divided into two parts. An objective test called for responses to multiple-choice questions about grammar and writing structure; the second part of the test required the students to write an essay on a given topic.

In those English courses in which TICCIT was used as it was intended to be used, as a mainline instructional source, there were consistent and statistically significant differences favoring TICCIT over lecture sections, on the objective test of writing skills. Student performance on essay tests was also better in TICCIT sections, except in a situation in which a part-time instructor unfamiliar with TICCIT conducted the TICCIT class.

On the other hand, when TICCIT was not used as the primary source of instruction, achievement effects were mixed. In those classes using TICCIT adjunctively, only students of relatively high ability experienced gains on the posttests. Those TICCIT students of low entrance ability scored lower on posttests than similar students in lecture sections. Use of the system merely as a supplement to classroom instruction simply does not yield the same consistency of positive gains found with the mainline applications in both English and math.

At BYU, there were also two measures of achievement. All TICCIT and non-TICCIT students were given a departmental objective test of writing skills and a university-administered writing competence exam of objective and essay skills.

There were no significant differences in the scores on the departmental exam. However, the exam had not been carefully designed to take into account the differences in the TICCIT and non-TICCIT course outlines as had the ETS exams. The departmental exam had been primarily aimed at the non-TICCIT courses.

For essay tests and for overall pass/fail ratio on the university-administered English composition exam, TICCIT students did significantly better than non-TICCIT students. Over a three semester period, BYU tested 160 TICCIT students and 137 non-TICCIT students who had had the same instructor. Approximately 65% of the TICCIT students passed the exam compared to only 56% of the non-TICCIT students, primarily due to the TICCIT students achieving higher scores on the essay component of the exam.

Summary. TICCIT exerts a significant and positive impact upon student achievement. For five out of the six observed math courses significant positive results occurred, often resulting in TICCIT posttest scores at least ten percent higher than lecture, corresponding to two-thirds of a standard deviation. In addition to overall higher achievement on posttests, TICCIT promoted higher achievement in the ability of students to solve routine problems and was equal to lecture in promoting adequate performance in the most advanced content areas of the math curriculum. In four of the six math courses, students in TICCIT courses scored about ten percent higher on topical tests of immediate learning than similar lecture students. In English, results for courses comparing TICCIT as designed and lecture likewise demonstrated significant positive effects for TICCIT students, resulting in an estimated difference of five percent in scores on objective tests of writing skills. Except where inexperienced instructors affected results, TICCIT students also scored higher in essay tests.

COURSE COMPLETION RATES

The percentage of students completing TICCIT courses within an academic term varied tremendously among the various applications of the system. The most reasonable explanation for the variations appears to be the environment in which the TICCIT student learns. In existing literature, it is fairly clear that without some type of external management effort, fewer students will complete an individually-paced course than an instructor-paced course (Atkins and Lockhart, 1976; Coldeway and Keys, 1976; Coldeway, 1974; Lloyd, McMullin, and Fox, 1974, Coldeway and Schiller, 1974).

In the traditional classroom situation, motivation to complete the course within a given time frame is to some degree built-in, usually in the form of scheduled assignments and tests. To supply the motivation to complete the course on schedule in an otherwise self-paced TICCIT environment, the instructor must adopt the role of manager/advisor, providing individual students with progress and achievement goals and evaluating each student's performance in relation to those goals. Completion rates in TICCIT courses will depend upon the extent to which an instructor adopts that role.

In the original community college implementation, there was no emphasis upon completion as a program outcome; however, early in the demonstration it became apparent that high completion rates would not be achieved without a more structured environment. Therefore, the TICCIT users at BYU established the use of TICCIT in environments which included a weekly scheduled class meeting, small group discussions, individual student/teacher meetings, scheduled quizzes and other features designed to motivate the student to complete within the academic term.

This model produced completion rates generally higher than in comparable lecture sections at BYU. In math, the average completion rates for classroom instruction was 65%. Classes taught by one of the TICCIT math authors averaged higher rates than classroom rates. Two classes taught by teachers previously unfamiliar with TICCIT were mixed; one had a completion rate of 72%, the other 33%. Rates for TICCIT English classes were very favorable, ranging from 88% to 96%. Those rates exceeded the very liberally-calculated completion rate of 82% for the classroom (the lecture sections did not include in the rate calculation any of a large number of students withdrawing before the third week of class, as did the TICCIT sections). Clearly, when efforts are expended to obtain high completion rates, it is possible to do so in an environment which attempts to pace students within certain time periods. Although the community college implementations of TICCIT did not generally include such concerted efforts to increase completion rates, the TICCIT English instructors were generally very imaginative in their approaches to directing the TICCIT classes, using features such as class meetings and group discussions. It was apparent that those efforts did result in completion rates that, although lower than lecture classes at the community college, were significantly higher than those in math classes in which there was basically no effort to pace student learning according to any schedule. In three of the five cases in English there were significant negative impacts upon completion attributable to the TICCIT condition, but in every case the English completion rates were substantially higher than found in the TICCIT math sections. The rates for English were generally 10% to 20% lower than classroom completion rates, which ranged from 60% to 75%.

The math courses at both community colleges were generally implemented without any of the tools discussed above that would tend to motivate pacing within the academic term. Total self-pacing was the rule for math classes; there was no penalty for re-enrolling for the same course in subsequent terms. There appeared to be little or no discussion with individual students concerning learning pace, and there were no group or class discussions. The results of the analysis do show significant treatment effects upon completion. In 10 of the 12 math classes analyzed, significant (at $p < .001$ levels in all but one case) and negative treatment effects were associated with the TICCIT sections; the two remaining sections showed no significant treatment effects. The rates for TICCIT sections were generally about half the size of lecture rates, which were about 40% to 55%.

Implementation of more stringent management efforts at the community colleges since the demonstration has resulted in substantial increases in the TICCIT completion rates. One college has reported that completion rates for math have more than doubled and are approaching the rates in the lecture sections. The latest figure from the other community college shows a completion rate in TICCIT English (72%) which surpassed the rate in the lecture sections.

COMMENTS

Those who continue to develop and market the TICCIT system have been very pleased with TICCIT evaluation results. The achievement data consistently demonstrate significant and substantial positive gains for TICCIT students in both math and English. The measure of success for TICCIT was to be its impact upon achievement compared to traditional instructional methods. In that light, it must be said that even in its initial operation, TICCIT was highly successful, teaching math and English students better than was done in traditional classes.

Perhaps the most gratifying results were found in the English composition courses. Efforts to develop computer-assisted materials for composition instruction were met with constant skepticism. Yet, the evaluations have shown that completion rates in TICCIT English classes have been similar to or surpassed completion rates in lecture classes. More important was the finding that those students completing TICCIT scored consistently higher on objective and essay posttests.

Similarly, students in TICCIT math classes performed consistently better on course posttests and on topical tests than their counterparts in lecture sections. Although completion rates for the highly managed environment did surpass lecture rates, the rates for TICCIT math classes within the evaluation context were disappointingly low. Completion was not an outcome highly emphasized during the demonstration, but the higher completion rates found in the BYU study would have been just as desirable in the community college setting.

A very important question for those involved in current development of TICCIT and CAI in general is why completion rates in math were so much lower at the community colleges than the English rates, especially since the two courses were basically identical in design. It has already been suggested that the difference lies chiefly in the degree of instructor involvement in managing student progress. Other contributing factors are possible and have been examined and discussed below.

The explanation apparently favored by ETS was that the TICCIT system itself may favor students of high ability to the detriment of low ability students. While the completion rate figures for the math courses do appear related to entrance ability, the figures relate to completion within a single term. Such results relate more to the speed of acquiring mastery of the objectives rather than the ability to master the objective at all. This consideration and others which are discussed below tend to dispel the notion that TICCIT is suitable only to high ability students.

The ETS conclusion concerning ability is based upon the observation of differences in pretest means of students finishing the course in TICCIT sections and those students finishing the lecture sections. The differences in the fall term were particularly large. However, in later terms the differences in pretest mean scores became much smaller, primarily because of the addition in later terms of the scores of students taking more than one term to complete. The effect of this can be seen in the analysis of the fall term of beginning algebra at Phoenix. The pretest mean for students finishing TICCIT was 44.86 while for lecture it was 41.28. Logically, one might conclude that any difference in posttest scores (actual scores for TICCIT and lecture were 40.06 and 38.82, respectively) must be due to the high entrance ability. However, further analysis limiting the comparison to higher ability students (defined in this case by pretest scores greater than or equal to 35) still showed positive achievement effects. Later analysis eliminating the constraint of completions within a term included a sizeable group of students who had completed TICCIT math during the spring semester. The mean pretest score of this group was 33.92, which is below the earlier defined high ability level of 35, below the pretest mean of lecture students completing the fall semester (38.64), below even the mean of the lecture students from the fall who did not finish within one term but also completed in the spring (34.62), and below the mean pretest scores of all students taking the pretest. Yet this broadening of the applicable entrance ability did not subtract from the overall achievement result. In fact, as mentioned earlier, the inclusion of all the students requiring more than one term to complete increased the impact of achievement results. Although the view of the TICCIT completion and achievment results within a single academic term appears to support an argument for TICCIT suitability only to high ability students, the results across terms weakens the argument with the inclusion of substantial numbers of students with much lower entrance abilities who, nonetheless, add a greater degree of significance to the positive achievement data. It does appear that students with higher pre-test scores are able to finish TICCIT courses faster than those with lower scores, a finding not unusual with individually-paced learning systems.

An additional factor which weakens the high entrance capability argument is the topical test analysis. For topical testing, an even broader segment of the student population was included. Unfortunately, no detailed analysis is made of students completing topical tests in terms of entrance ability. The comments in the evaluation imply that the group taking the pretest was more inclusive than the group completing TICCIT but did not include the entire population beginning the course. The students were characterized as those able to make progress towards completion on TICCIT. Again with the enlarged population, there are still substantial achievement gains for TICCIT students in five of the six courses. Although the data is not clear, the implication from the topical test results is that the entrance ability argument must be still further weakened.

The ETS report's treatment of the topical tests seems to concern itself with the fact that the TICCIT population taking topical tests was still smaller than the population originally enrolled in the course. That fact may in some situations be compelling evidence in support of the contention that TICCIT is better suited to highly capable learners. But the context of that evaluation is a comparison of TICCIT and traditional instruction; therefore, the statement that TICCIT would be suitable only for high ability students implies that the lecture condition has a more positive impact upon those of lower ability. It is clear from ETS data, however, that the traditional instructional method also exerts a strong bias against the lower ability student. The issue then is not simply whether TICCIT exerts a bias towards low ability students but whether that bias is significantly greater than that already existent in traditional instruction. Without very substantial differences between any bias exerted by TICCIT and bias exerted by lecture methods, the statement that TICCIT is suitable for high ability students could not be justified. On the basis of the results of across-term analysis and topical testing discussed above, there does not appear to be such differences to justify the conclusion that TICCIT applications should be limited to those involving only highly capable learners.

Further arguments against the general hypothesis that TICCIT is not suitable for students of less than high ability are found in the following observations. First of all, the math courses under consideration were fundamental courses in community colleges with very liberal enrollment policies. A large number of the classes for which data were taken did not qualify as college credit courses but were developmental classes to compensate for pre-college deficiencies. They are necessary and important courses, but it may be difficult to support any general statement concerning "high ability" students on the basis of observations only from these courses. Perhaps even more interesting is the relationship between two of the courses at Phoenix. Generally, students enroll in Math 7 only because they have not demonstrated the entrance ability, as judged by performance on a pretest, required for Math 106. Be definition, students in Math 7 would be of lower entrance ability (i.e. lower pretest scores) than those in 106. Yet the analysis for Math 7 demonstrated one of the strongest positive effects for TICCIT students.

In short, there appears to be satisfying evidence that TICCIT can be successfully applied in learning environments in which there are varied entry abilities. Although entrance ability must be considered when implementing TICCIT, the consideration should not preclude those of low entry levels. TICCIT was built on a mastery hypothesis; that is, given adquate instruction, defined as having a complete learning hierarchy, properly stated rules with alternate versions of the rule, and extensive files of examples and practice problems with helps, any student with the prerequisites for a given lesson could, given enough time, pass that lesson. The TICCIT math course assumed a minimum set of prerequisites for entering students. The effect of not having the prerequisites was more pronounced in math than in English since math is a cumulative subject. Large numbers of the community college students did not possess the minimum prerequisites required by TICCIT, yet they were allowed into the TICCIT courses as they were allowed into the lecture courses. The design philosophy of TICCIT does not specify that all students who lack the prerequisites would be able to succeed--no matter how much practice on a higher order math skill they were able to engage in. Rather, the TICCIT design hypothesis is that for such students, other TICCIT lessons could be developed which would give them the prerequisites so that they would be able in a manner most efficient for their particular learning rate and aptitudes to complete the higher lessons.

Evidence that the students did lack the necessary prerequisites is found in the large numbers of students who were not able to complete the first TICCIT unit or who took an extremely long period of time to do so. Once students were able to complete the first unit in math, chances for their successful completion of the course were greatly improved. However, it cannot be said that the TICCIT system itself is inadequate if students not possessing the presumed prerequisites have difficulty achieving mastery on higher order lessons. If the ETS data is viewed as formative evaluation data, the data says that the hierarchy must be extended downward and that existing material should be simplified. That the model is less effective for less able students does not seem to be a valid conclusion.

The entrance ability hypothesis does not seem to explain adequately the low completion rates in the community college math courses. Comparing the completion rates of math to those of the self-paced programmed instruction makes it reasonable to assume that the problem of low completion rates is a generic problem with self-managed instruction. The rates found in TICCIT were comparable to the rates in the self-paced programmed instruction at Alexandria and to the historical rates of similar programs at Phoenix. The evaluation record mentions the problem a number of times, referring, for example, to the fact that many students failed to attend classes regularly, or at all, when pressures from other non-self-paced courses increased. Records of the number of hours spent on-line also indicated that students not completing the course spend fewer hours on the system than the number of hours assigned for classroom usage in the lecture section. (For example, a study conducted by Alexandria indicated that average usage for students not completing TICCIT sections was approximately 30 hours. The lecture sections met for 50 hours per term.) There is apparently a problem with the management of student progress which greatly contributes to lower completion rates.

The management problem is not surprising, however. From the beginning of the project, TICCIT developers understood that such problems existed with self-paced instruction and did attempt to create solutions in advance. The TICCIT math and English courses were both designed to be implemented in an environment in which the instructor took an active role in the management of student progress. The originally defined role of manager-advisor to be assumed by instructors included goal-setting and performance evaluation behaviors. Such a model was implemented much more fully in the English than in the math courses. As has been seen, completion rates in English were higher; in fact, the lowest completion rate in any English course was higher than that for any math section.

The English and math course designs themselves were simply not divergent enough to result in such large completion differences. However, a review of the implementation contexts show striking differences. ETS observed that English instructors took an active role in prescribing study materials, conducting small group discussions about common difficulties, and handling individual student problems. On the other hand, math students had much less instructor contact, and the great majority of contact that did occur was centered on content problems rather than individual progress problems. At Phoenix, students were not even assigned to particular math instructors; consequently, the instructors assigned to the lab in a given hour may not have been able to identify individual students. In short, there appear to have been substantial differences between both the quantity and quality of instructor interaction with math and English students.

It is perhaps understandable that better management efforts were not implemented. The developers did attempt to provide features in the system, such as various reports concerning individual student progress, but for various reasons no systematic method of handling the management problem was provided. It is to the credit of the colleges themselves that programs were developed in the English courses that did improve student completion rates through better management. The methods implemented, including small group discussions and closer involvement in individual assignments, could and should also be used in mathematics. The possible impact of management was not discussed in the ETS evaluation of math completion rates, possibly because none of the math sections dealt with the problem and because no obvious solutions to the problem were ever suggested or implemented. But because of the basic similarities between the courses, there is reason to believe that similar management efforts would also increase math completion rates. The evaluation did indicate a strong correlation between time spent on-line and the number of math lessons completed. To improve the rate of completion, it would appear then that measures are needed to improve student motivation to spend an adequate amount of time on the system.

Further evidence for the efficacy of management efforts is found in the BYU math implementation, discussed earlier in the completion rates section, in which TICCIT completion rates did exceed those of the classroom. The literature would also support the hypothesis that increased management efforts will result in higher completion rates (Lloyd, et. al., 1974; Miller, Waver and Semb, 1974; Malott and Svinicki, 1969).

The developers were highly pleased with the results of the evaluation. Very clearly positive achievement results for instruction in English was a result expected by very few people. Yet the results do show that grammar and composition can be taught systematically with positive effects on student writing capabilities. Even in the initial year of implementation of a radically new instructional method, nearly as many students finished the course with TICCIT as with the established instructional method and performed significantly better on achievement tests. As instructors have become experienced with the system, its effectiveness continues to be manifested: completion rates now often exceed those of the classroom. Achievement for math students on TICCIT is equally pleasing. Distracting slightly from that enthusiasm were the lower completion rates found with the math sections within the ETS evaluation. However, the developers find no reason to believe that there is any inherent characteristic of TICCIT instruction that would prohibit completion in math from being equal to that found with English. There is ample evidence, including already improved completion rates, that the management techniques used to improve completion for English classes are just as effective in promoting completion rates in math comparable to or exceeding those in lecture classes. TICCIT is a very powerful instructional tool, one which should continue to have a great impact on the future of educational practices.

REFERENCES

1. Atkins, J.A. and Lockhart,L. Flexible vs. Instructor-paced College Quizzing: A Behavioral Analysis of Preference and Performance. In L.E. Fraley and E.Z. Vargas (Eds.), Behavior Research and Technology in Higher Education.
2. Bunderson, C. Victor. A Rejoinder to the ETS Evaluation of TICCIT. CTRC Technical Report No. 22. Provo, Utah: Brigham Young University, 1977.
3. Coldeway, D.O. and Keys, C.B. The Effect of Instructor-Pacing on Later Self-Pacing. In L.E. Fraley and E.A. Vargas (Eds.), Behavior Research and Technology in Higher Education. Gainesville, Fla.: Society for Behavioral Technology and Engineering. Psychology Dept., University of Florida, 1976.
4. Coldeway, D.C. Comparison of small-group contingency management with the personalized system of instruction and the lecture system. In J.M. Johnston (Ed.), Research and Technology in College and University Teaching (proceedings, second national conference, Atlanta, 1974). Gainesville, Fla.: Society for Behavioral Technology and Engineering, Psychology Dept., Univ. of Florida, 1975.
5. Coldeway, D.C. and Schiller, W.J. Training proctors for the personalized system of instruction. In R. Ruskin and S. Bono (Eds.), Personalized Instruction in Higher Education. Washington, D.C.: Center for Personalized Instruction, Georgetown University, 1974, 111-118.
6. Evaluation of the TICCIT Computer-Assisted Instructional System at the Community College (Draft). National Science Foundation Contract No. NSF-C731. Princeton, N.J.: Educational Testing Service, 1977.
7. Lloyd, K.E., McMullin, W.E. and Fox, R. Rate of completing unit tests as a function of student-pacing and instruction-pacing. Paper presented at the Fifth Annual Conference on Behavior Analysis in Education, Kansas City, October, 1974.
8. Malott, R.W. and Svinicki, J.G. Contingency management in an introductory psychology course for one thousand students. Psychological Record, 1969, 19, 545-556.
9. Miller, L.D., Waver, F.H., and Semb, G.A. A procedure for maintaining student progress in a personalized university course. Journal of Applied Behavior Analysis, 1974, 7,87-91.
10. Sasscer, Monica Flynn, TICCIT Project 1976-77 Final Report. Alexandria, Va.: Northern Virginia Community College, 1977.

TABLE 1

ACHIEVEMENT SUMMARY
MATH
Academic Year 1975-76

							Pretest-Posttest Observations			
							Pretest		Posttest	
Course	Term	Comparison Group	All N	Pretests Mean	All N	Posttests Mean	N	Mean	N	Mean
Phoenix										
007	Fall	Total	415	37.26	182	39.42	152	41.80	152	39.32
		TICCIT	161	35.32	27	44.26	22	44.86	22	43.32
		Lecture	254	38.49	155	38.58	130	41.28	130	38.64
	Spring	Total	332	36.24	181	38.96	172	39.05	172	38.78
		TICCIT	110	36.15	35	40.57	32	40.06	32	40.09
		Lecture	222	36.28	146	38.58	140	38.82	140	38.49
Alexandria										
31	Fall	Total	195	31.72	58	38.90	52	38.19	52	39.27
		TICCIT	90	31.56	9	45.33	9	47.22	9	45.33
		Programmed	47	29.74	7	34.86	4	36.50	4	34.75
		Lecture	58	33.59	42	38.19	39	36.28	39	38.33
	Winter	Total	105	30.00	62	38.53	46	34.09	46	39.07
		TICCIT	41	27.29	25	40.00	21	34.33	21	40.86
		Programmed	33	32.21	16	38.56	10	34.00	10	39.10
		Lecture	31	31.23	21	36.76	15	33.80	15	36.53
	Spring	Total	130	29.01	38	38.83	32	38.37	32	38.56
		TICCIT	41	30.12	11	40.36	9	41.33	9	40.56
		Programmed	58	26.14	16	36.44	14	33.21	14	35.79
		Lecture	31	32.90	11	40.73	9	43.33	9	40.89

TABLE 2

ABILITY LEVEL

Factual recall

Manipulation

Solution of routine problems

Demonstration of concept comprehension

CONTENT CATEGORY

1. Arithmetic
2. Algebra
   - (a) Simplifying, including properties of and computations with signed numbers and complex numbers
   - (b) Equations, including substitutions, equivalent equations, solution of equations, and word problems.
   - (c) Exponents, radicals, and logarithms
   - (d) Factoring, multiplying, and dividing algebraic expressions including quadratics
   - (e) Number line and coordinate plane
   - (f) Inequalities, absolute value, and sets including solution sets
   - (g) Sequences, arithmetic and geometric progressions
   - (h) Permutations and combinations, binomial theorm

TABLE 3

ACHIEVEMENT RESULTS: ANALYSIS FOR IMMEDIATE LEARNING
Alexandria and Phoneix
Math Courses
Academic Year 1975-76

Course	Term	Topical Test	Number of Students	Multiple Correlation	Independent Variables	Regression Weights	Value of t-statistic
007	Spring Semester	Linear Equations	72	.71	TICCIT (1) Lecture (0)	2.76	3.16**
108	Spring Semester	Algebraic Expressions	47	.60	TICCIT (1) Lecture (0)	2.27	1.42
31	Fall Quarter	Linear Equations	63	.62	TICCIT (1) Lecture (0)	.41	.35
31	Winter Quarter	Linear Equations	47	.63	TICCIT (1) Lecture (0)	2.17	1.83
31	Spring Quarter	Linear Equations	27	.75	TICCIT (1) Lecture (0)	2.61	1.61
32	Spring Quarter	Algebraic Expressions	52	.30	TICCIT (1) Lecture (0)	-2.09	-1.61

**p < .01

STUDENT CONDUCTED SURVEY OF THE OPINIONS OF ORANGEBURG RETAIL BUSINESS MANAGERS, WHITE AND NON-WHITE, ON LOCAL POLITICAL ISSUES

by

Robert J. Kirsch
Claflin College

This paper presents an analytical description of a teaching experiment on the undergraduate level. The project involved: (1) the directed design, by the students, of a mini-questionnaire (eight questions, each with a maximum of four answer choices) to be compatible with DIGITAL'S canned program, SAP; (2) the random selection and questioning of Orangeburg retail business managers; (3) compilation of the punched data by the campus PDP/8M; (4) classroom discussion and interpretation of the aggregate data; and (5) written student analysis of survey results and evaluation of the project.

This paper presents an analytical description of a teaching experiment on the undergraduate level which combined survey research methods, as applied to the study of political behavior, with computerized compilation of the collected data.

The experiment was conducted in Orangeburg, South Carolina, a small, southern town (approximately 17,000) whose population is racially mixed. Blacks hold a slight numerical majority over Whites. The computer facilities of Claflin College, a small (900 students), predominantly Black, church-related (Methodist) college, were utilized. Participants in the project were eight Claflin students, mostly juniors, enrolled in Principles of Political Behavior, offered in the spring semester, 1977, and their instructor, the author.

The project involved five steps: (1) the directed design, by the students, of the survey questionnaire; (2) the random selection and questioning of interviewees from among the Orangeburg retail business managers, White and Black; (3) compilation of the punched data by the campus computer, a PDP/8M; (4) classroom discussion and interpretation of the aggregate data; and (5) written student analysis of survey results and evaluation of the project.

Designing the Questionnaire

In preparing for the course, I had acquired DIGITAL Equipment Corporation's canned-program, Huntington II Simulation Program--SAP. SAP is an acronym for Survey Analysis Package. This program does statistical calculations useful for examining and analyzing survey data. It is designed to be used to facilitate a student-conducted survey research project. The program does the following:

1) Compute the mean and standard deviation for any variable. (With OPTIONS 1 through 5 a Recode OPTION is included.)

2) Construct a table of observed frequencies for any two variables, compute chi square, degrees of freedom (df), and allow the user to calculate the correlation coefficient gamma if desired.

3) Construct a table of observed frequencies by row percentages for any two variables.

4) Construct a table of observed frequencies by column percentages for any two variables.

5) Construct a table of expected frequencies for any two variables.

6) Stop the program.[1]

The amount of raw data enterable into the program depends upon the storage capacity of the particular computer. The number of questions that can be stored for each respondent also depends upon computer storage capacity.

SAP is well documented by the vendor. Instructional materials include: (1) a "Teacher's Guide"; (2) "Student Workbooks"; and (3) a "Resource Handbook." Careful reading of the three should enable anyone minimally familiar with computer techniques and survey research methods to employ successfully the SAP program. It is possible, given sufficient machine capacity, to construct a compatible questionnaire consisting of a maximum of eight questions/variables and a variable range of from one to eight.

Following partial pretesting, I concluded that with the campus computer I could utilize a questionnaire of eight questions, each with a maximum of four answer choices. I was now ready to direct my students in the design of the project questionnaire.

To prepare my students for the task, I required them to read the following three items:

1) Robert J. Kirsch and Carl E. Swenson, "Student Survey: A Political Science and Computer Science Team Teaching Experiment," Proceedings of 1976 Conference On Computers In the Undergraduate Curricula, CCUC/7, SUNY, Binghamton, June 14-16, 1976, pp. 301-305.

2) Robert J. Kirsch- "Socio-Political Profile of Claflin College Students," Claflin College Review, v. 1 (no. 1), pp. 52-60.

3) Herbert McClosky, Political Inquiry: The Nature and Uses of Survey Research (N.Y.: Macmillan, c1969), Chapter I.

Following classroom discussion of the above, each student was assigned responsibility for drafting five questions to cover one of the following: local zoning, utilization of local taxes, police protection, local law enforcement, local governmental officials, such as the mayor and city councilmen, public services, such as utilities, water and sewage, construction and maintenance of roads and public parking facilities; and provision of health and emergency services. I hoped to be able to average one usable question per student from the five submitted by each.

I coached the students in how to draft questions which would be compatible with the SAP canned-program. I informed them that they would have one week to complete and submit their questions.

Upon receipt of the questions, I carefully edited them and chose those most appropriate. I constructed a preliminary questionnaire which was randomly distributed by my students among their fellow students. The purpose was to test it for clarity of language. After testing, the questionnaire was revised, retyped, and run-off for distribution. While there were obvious questions about the cohesiveness of a questionnaire so constructed, I felt that the benefits derived by the students from writing it themselves far outweighed the possible drawbacks.

The final questionnaire was designed so that the first two questions could serve as keys for evaluation of the responses. The first two questions and their possible responses were:

1. I am

    1. white male 2. non-white male
    3. white female 4. non-white female

2. I am

    1. under 35 2. 36-45
    3. 46-60 4. 61 and over

I had hoped thereby to be able to code responses on the basis of race/sex and age. Due to a bug in the program, however, this was not possible. The computer tabulations of respondents' ages were incorrect by a factor of ten. The bug in the program resulted in an addition of five to the number of responses to "1. under 35," and five to "2. 36-45." This program error was later transformed to all tables involving a cross tabulation with age responses. While it would be possible to debug the program, this error should have been caught by the vendor before marketing.

Random Selection and Questioning of Interviewees

Once the final questionnaire was prepared, one student volunteered to procure a map of the City of Orangeburg. Each student then chose a letter/number-coded part of the city in which to conduct the interviews. Students who had access to an automobile took the more distant points; others took those within walking distance of the college. Each student was asked to interview fifteen local merchants, White and Black. It was decided that every third retail business manager would be contacted until the student had interviewed his/her fifteen. Students were given three weeks to complete the interviewing.

Due to the limited scope of the questionnaire it was not possible to make distinctions between owner-managers and salaried-managers, or between Orangeburg city residents and non-residents. To secure such information it would have been necessary to limit the number of politically relevant questions.

Computer Compilation of the Data

Local merchants were very cooperative with my students. Of 120 contacted, 118 completed and returned the questionnaires after inserting them in the envelopes provided and sealing them. This they did either while the students waited or before the students returned for them.

Once the questionnaires were returned, the students removed them from the envelopes and transferred the responses to the Data Coding Forms the vendor provided in each Student Workbook. Once encoded, the students took turns punching the data on paper tape at the teletype terminal.

There were some minor difficulties encountered in tabulation of the data due to the small capacity of Claflin's PDP/8M at the time. (The campus system subsequently has been expanded to a 32K core.) It was necessary to split the data into three parts keyed to the first two questions and to tabulate each separately. While cumbersome, this method was still much quicker than tabulation by hand. A further drawback was that the canned-program was much too complex to allow students with minimal exposure to computers and statistics to have "hands-on" experience. The instructor input the data and instructions.

Once the data were output, it was necessary to label by hand each table. This proved to be confusing at first for the students. However, once I had demonstrated how it should be done, they grasped the method. In the future, it would be most helpful if the vendor could so modify the SAP program to permit labeling at the time of data input.

Classroom Discussion and Interpretation of the Data

Following labeling of the data, I divided the class into three groups of three apiece. Since there were only eight students, I served as the odd man on the third team. Each group was instructed to analyze its segment of the questionnaire. Group I was responsible for questions 1, 2, 3, 4; group II for 1, 2, 5, 6; and group III for 1, 2, 7, 8. Each group was to select a spokesman who would present the results of the group's analysis to the entire class. Each group received three Spirit Duplicator Carbons upon which to type its findings. These were then reproduced and distributed to the class. Each presentation was followed by open discussion and free criticism of observations made. There were noticeable increases in acuteness of analysis with each subsequent report.

A number of telling remarks were made by the different group spokesmen. Noting that white merchants were disproportionately more numerous than nonwhite, especially in upper age groups, the spokeswoman for group I accounted for it by noting that, in the past, "Blacks got into jobs that offered security and stayed with these rather than try to break into business where they were unwanted."

The spokeswoman for group II observed that the attitudes of Whites and Blacks toward police protection in Orangeburg County differed. She attributed white attitudes to the fact that "Whites headed the police force...(They) really believe that it is good." Nonwhite attitudes might derive from the fact that "the police force is not protecting their rights as it would (those of) Whites."

The spokesman of group III commented on the limitations of the data which "could not tell us the importance of the merchants' attitudes towards the subject...We could not tell how well they knew the subject....(It was not clear) how long the persons held their attitudes...nor was it possible to verify truthfulness."

The one drawback to assigning separate groups responsibility for analysis of each segment was that the workload tended to fall disproportionately upon the group spokesman. The intention was to encourage student discussion and group analysis of the data. This did not occur to the desired degree.

Written Student Analyses and Project Evaluations

Upon completion of the classroom discussion of the survey results, students received written guidelines for the preparation of their final written analyses and project evaluations. The guidelines stipulated:

Each paper should consist of two parts: (1) an analysis of survey results; and (2) an evaluation of the entire project.

Each part should be between 5 and 6 pages in length. The analysis part should consist of an evaluation and interpretation of the data and an indication of its significance.

The evaluation part should begin with a brief opening statement outlining the project.

This should be followed by a critical evaluation of the project in which the student indicates the following:

1) The project's effectiveness as a teaching device;
2) The project's significance as a learning experience;
3) Student cooperation and preparedness levels;
4) The merits of the simulation for teaching survey research methods;
5) The project's relevance and effectiveness in comparison with more traditional teaching methods; and
6) Other relevant observations--positive and/or negative--which the student believes worth making.

There should be a brief concluding statement.

Each student should prepare his/her own paper in his/her own words. The student's paper should be neat, orderly, and grammatically correct.

Consultation with other students before preparation of the paper is recommended. However, each student must exercise great caution to ensure that other's opinions do not unduely influence his/her own evaluation.

Individual student analyses of survey results tended to reflect the thrust of the in-class discussion of same. There was greater awareness of the implications of the data; there were fewer misinterpretations.

Among the more interesting aspects of the students' analyses were their accounts of their experiences while distributing the questionnaires. One student (an African) commented:

> ...the non-whites that I interviewed had a funny...attitude towards the Mayor...Some...said they have not experienced any change (in) Orangeburg. And they do not expect to see any in their lifetime...
>
> ...rating Orangeburg County's Ambulance Service, they (stated) they cared for their 'ride'--car--rather than the Ambulance. I wish I would have gotten a precise explanation for their negativism in this matter. I tried hard to get some vital information, but my efforts fell flat.

Levels of student participation and enthusiasm were high. In general, the class regarded the experience as more valuable than more traditional assignments, such as term papers and oral reports.

The project was deemed a success because: (1) it encouraged direct student involvement; (2) it exposed the students to practical applications of social science methods; and (3) it led to student interaction with area merchants and heightened their confidence in their ability to deal with similar contacts in the future.

Footnote

1. Daniel Klassen, Huntington II Simulation Program--SAP: Teacher's Guide, Digital Equipment Corporation, c1973, p. 2.

COMPUTER-AIDED REVIEW LESSONS IN ENGLISH GRAMMAR

Edward A. Kline
University of Notre Dame

Against a background reflecting the failure of college students to master and to understand grammatical skills, the author reports the design and implementation of fifty-seven review modules distributed over the areas of English phonology, morphology, syntax, spelling, and punctuation. Following excerpts from a review module, discussion turns to the advantages of the programs written for the time-sharing mode of computing in both the VS.BASIC and FORTRAN languages. The paper concludes with the results of an evaluation by 211 students who used the programs in the Fall Semester of 1976 at the University of Notre Dame.

Students matriculating into higher education begin their studies, with few exceptions, in required courses devoted to mastering the skills and the art of effective communication in English prose. Toward these ends, writing courses stress rhetorical principles and properly so, while very few courses include any study of grammar. Whereas the formal study of rhetoric is a new discipline to most of these students, the formal study of grammar usually is not. The roles of grammar and rhetoric in achieving the desired purpose can be compared to an athletic contest wherein grammar becomes the rules and conventions of the game while rhetoric represents the actual choice of plays and maneuvers that permits an individual to score points. Mastery of the grammar or rules alone will not result in successful play, and knowledge of the rhetoric of the game, without an understanding of the grammar involved, results in inane play. While no athletic coach, real or imaginary, could conceive of winning a game without a mastery of its "grammar," many teachers of writing and most of their students actually believe, or act as if they believe, that such is, indeed, the case. It was against such a background that computer-aided review lessons in English grammar were developed at the University of Notre Dame in 1975. Many students were simply not scoring points.

Incoming university students display varying degrees of mastery of the rudiments of English grammar ranging from fully proficient on the one hand to a grammar school level on the other with all the variations between these extremes adequately represented. The causes of grammatical ignorance and uncertainty are undoubtedly manifold and have received adequate attention from professionals in the discipline as well as school administrators and irate parents. It is not the purpose of this paper to explore the causes but to discuss remedial measures after the fact.[1] Whatever the causes may be, current college students are afflicted. And the malady does not rest with only the freshmen: upper classmen, having some training in rhetorical principles, also struggle to communicate in prose. Despite a maturation in their reasoning processes and a sincere desire to write well, these advanced students attempt to maneuver through a game in which they are uncertain of the rules.

The computer-aided review lessons in English grammar are designed to aid such students. Written in the VS.BASIC machine language, and also available in a FORTRAN version, the program format is tripartite. Part I provides for the construction of the review modules which contain the exposition of various grammatical and spelling principles followed by appropriate examples. This leads to a series of multiple choice questions to which the student responds. Part II of the program controls student access of the system between the terminal and the computer, and Part III contains the editing functions which permit the instructor to add, delete, change the order of questions and foils, etc.[2]

The fifty-seven review modules contained in Part I are categorized under the headings of morphology-syntax, punctuation, spelling, and phonology. The latter category is primarily designed for students whose native language is not English and those students whose speech reveals dialectal variations, but it has also proven very useful in introductory linguistics courses. A listing of the module codes, the number of questions contained therein, and their description follows:

CODE INDEX AND DESCRIPTION OF ENGLISH CAI LESSONS

	CODE	QUESTIONS	DESCRIPTION
			MORPHOLOGY-SYNTAX:
1.	ANTEPRO	6	Antecedents; Pronoun Reference
2.	APPOSIT	5	Appositives
3.	COMPMOD	9	Comparison of Modifiers
4.	EXPLETIV	5	Expletive Patterns
5.	NOMOBJPR	4	Pronouns: Nominative, Objective
6.	NOUNMOD	7	Modified Noun Elements
7.	PREPHR	9	Prepositional Phrases
8.	PARALLEL	9	Parallel Structures
9.	PASSENT	4	Passive Sentence Patterns
10	PHRCLMOD	3	Phrasal and Clausal Modifiers
11.	SENTEN	8	Seven Basic Active Sentence Patterns
12.	SUBCLAUS	12	Subordinate Clauses
13.	SVAGREE	12	Subject-Verb Agreement
14	VERBALS	10	Verbals: Words and Phrases
15.	VERBFRMS	9	Verb Forms
16.	VERBMOD	3	Verb Modifiers
17.	VCLAUSE	5	Verbal Clauses
18.	VCONDAUX	5	Verbs: Conditional Auxiliaries
19.	VSUBJUNC	4	Verbs: the Subjunctive
20.	USAGEA	11	Usage Problems, I
21.	USAGEB	8	Usage Problems, II
21		148	

PUNCTUATION

	CODE	QUESTIONS	DESCRIPTION
1.	CAPITALS	8	Capitalization
2.	PUNCCOL	4	The Colon
3.	PUNCCOM	17	The Comma
4.	PUNCDASH	3	The Dash
5.	PUNCENDS	6	End Stops
6.	PUNCPABR	4	Parentheses and Brackets
7.	PUNCQM	4	Quotation Marks
8.	PUNCSEM	6	The Semicoln
8		52	

SPELLING

	CODE	QUESTIONS	DESCRIPTION
1.	ABIBLE	6	Suffixes: -able, -ible
2.	DBLC	7	Doubled Consonants
3.	FINALE	10	Final -e
4.	FINALY	5	Final -y
5.	GENNOUN	5	Genitive Nouns
6.	HOMONM	9	Homonyms
7.	IEOREI	7	Use of IE and EI
8.	LYALLY	7	Suffixes: -ly, -ally
9.	NPLURAL	6	Noun Plurals
10.	PREFIX	4	Common Prefixes
11.	SEED	3	SEDE-CEDE-CEED
12.	SPDIAG	19	Spelling Diagnostic
13.	SUFOUS	4	Suffix: -ous
13		81	

PHONOLOGY

	CODE	QUESTIONS	DESCRIPTION
1.	PRONUNAF	4	Affricate Consonants
2.	PRONUNCR	25	Review of Consonants
3.	PRONUNDI	10	Diphthongs
4.	PRONUNED	4	The -ed Suffix
5.	PRONUNES	12	The -es Suffix
6.	PRONUNFR	5	Fricative Consonants
7.	PRONUNNA	4	Nasal Consonants
8.	PRONUNLA	3	Lateral Consonants
9.	PRONUNSI	6	Sibilant Consonants
10.	PRONUNST	7	Stopped Consonants
11.	PRONUNSV	5	Semivowels
12.	PRONUNVO	10	Vowels
13.	PRONUNVR	20	Review of Vowels
14.	SSTRESS	8	Sentence Stress
15.	WSTRESS	4	Word Stress
15		127	
57	Modules	408	Questions

Under ideal conditions, the plan of implementation is uncomplicated. An instructor notes grammatical error patterns in the writing of a student, calls the fact to the student's attention, and refers the student to the appropriate section of a grammar handbook. After studying the handbook, the student activates the appropriate CAI program for immediate re-inforcement and testing. When a student selects an incorrect response to a question, the grammatical principle under study is reiterated before the student is requested to select another response. Re-inforcement also occurs when the student selects a correct response, since the answer may be the result of guesswork. Each of the modules is independent of the others, i.e. a student need not begin with the module numbered one and work sequentially to do remedial work in the module numbered ten. The descriptive approach utilized in the modules is structural in the phonology section and a blend of structural and traditional in the morphology-syntax sections, with the traditional approach dominant. Nevertheless, the terminological content of the modules is independent of any one grammar book, handbook, or linguistic approach. The jargon of the discipline is presented with its synonyms: the main or independent clause, the dependent or subordinate clause, the conjunctive adverb or connective adverb, etc.

Instruction on how to use a computer terminal, its telephone, the two input commands necessary to access the individual modules, and the one command to end a review session consume less than ten minutes. Instructions on how to respond to the questions are contained within the module:

> This program is a computer-managed review of a specific part of your course. Each question has five choices and your score will be computed based on your first response to each. After each question enter a number, 1 through 5, to indicate your response choice.
> Enter a 0 to receive a new question and enter a 10 to end the session.

The review module commences with a general overview of the grammatical or spelling principle involved and progresses to specific points followed by questions, as illustrated in the following excerpts: one from the module devoted to Antecedents and Pronoun Reference and another from the module, "The Comma":

> A pronoun usually has an antecedent which is a noun or another pronoun that has been implied or previously stated. The only pronouns that lack an antecedent are "I" and "you" and sometimes, an indefinite pronoun. (Consult your text if you are not sure of the indefinite pronouns.) The reference of a pronoun must always be clear in writing to avoid any confusion.
>
> A pronoun must agree in number (singular or plural) and in person (first, second, or third) with its antecedent. In the following sentences, which one contains faulty pronoun reference?
>
> 1. Every boy in the classes will have their chance.
> 2. Each of the boys will have his chance.
> 3. Everyone will have his chance.
> 4. No one will have his chance.
> 5. Somebody will have his chance.
>
> Enter your choice:
>
> 2. Incorrect. The antecedent of "his" is not "boys." "Boys" is the object of the preposition "of." Hence, "his" agrees in number with its singular antecedent "each." Try another answer.
> 3. Incorrect. "His" agrees in number with its sinular antecedent "everyone." Try another answer.
> 4. Incorrect. "His" agrees in number with its singular antecedent "no one." Try another answer.
> 5. Incorrect. "His" agrees in number with its singular antecedent "somebody." Try another answer.
> 1. Correct. "Their" is a plural pronoun. It does not agree in number with its singular antecedent "boy." It should read "his chance."

The purpose of adding comma punctuation to writing is to show which grammatical units are to be joined and which are to be separated. Various teachers may differ in their preferences for placing the comma, but the review questions which follow are satisfactory guides for any type of formal prose.

Independent (main) clauses or sentences joined by a coordinating conjunction (and, or, but, for, so, yet, nor) usually require a comma before the conjunction:

He went shopping downtown to buy a suit, but his sister went to the movies.

When a coordinating conjunction is not used to join two independent clauses or sentences, a semicolon is used:

He went downtown to buy a suit; his sister went to the movies.

There are two major exceptions to the rule of adding a comma before a coordinating conjunction:

A. Two very short independent clauses joined by a coordinating conjunction can omit the comma:
   We tried and we lost.
B. If one or both clauses already contains one or more commas and there is a possibility of confusion, a semicoln may be used:
   All gardening, whether it be of vegetables, shrubs, flowers, and especially roses, is fun; but it is also hard work!

Which of the following sentences contains an error in comma placement or omits a necessary comma?

1. John laughed but Mary didn't.
2. She intended to purchase apples, bananas, pears, and peaches, but she forgot to bring her money.
3. Did he bring the sandwiches to the picnic, or did he forget his promise to do so?
4. He received extra money for doing the job so well, so he bought a present for his mother.
5. She gave her daughter everything she wanted, yet the child was not happy.

Enter your choice.

1. Incorrect. Two very short independent clauses joined by a coordinating conjunction may omit the comma.
2. Correct. The first clause contains a series of necessary commas to separate the words in the series. Using a comma to separate the clauses may lead to confusion; hence, a semicolon is preferred after the word "peaches."

The advantage of this approach over a workbook or a programmed text are rather obvious. On the one hand, it eliminates the need for an instructor not only to devise daily work sheets but also to correct the sheets, and it eliminates the inevitable delay that ensues between the time a student hands in the work sheet and the instructor's return of it. On the other hand, this system provides instructional responses to correct and incorrect answers which few programmed texts do. Furthermore, the student can work through each module any number of times and the resultant output can be saved and used for study and reference purposes. Indeed, many students have worked through the modules more than once deliberately selecting incorrect answers to discover if their reasoning coincided with the response received when the incorrect answers were chosen.

A few other considerations are in order. The length of time to complete a module varies, of course, with the ability of the student user to answer the questions, but one-half hour constitutes the average time. To avoid monotony and to avoid lulling the student into passivity, the instructions for responding to the questions are varied: select the one correct sentence, select the one incorrect entry, which sentence below is incorrect? etc. The programs are equally adaptable to teletype and DECwriter terminals as well as CRT terminals; however, the use of the CRT terminal is not recommended unless it is also equipped with a printing unit.

In the Fall Semester of 1976, the review programs were first made available at the University of Notre Dame, albeit lacking the spelling modules, in conjunction with a freshman-level course, "Composition and Literature." In the forty sections of this course, it was agreed that use of the programs would be at the instructor's discretion. Ten faculty members opted for their use. It was further agreed that students would not be required to work through all or any of the review modules in these sections. An instructor would recommend a particular module when he or she noted grammatical or mechanical discrepancies in the student's writing. Both students and faculty were requested to evaluate the program at the end of the semester. Some responses appear in Appendix I, but a few are of special interest.

The evaluation device consisted of two sections, one devoted to the mechanics of terminal use and its attendant problems and one devoted to the lessons per se and their perceived practical value. In part one, the majority of the students responded that the verbal and graphic explanations on using a terminal were clear, that the LOGON and LOGOFF procedures provided no difficulties, and, as expected, noted mechanical malfunctions of older teletype terminals and waiting in line to obtain a terminal. In part two, a number of questions are indicative of student acceptance of the programs:

Did your instructor refer you to a handbook or grammar book before recommending that you do a particular CAI lesson?
Always: 118 Never: 53
Sometimes: 49 No Response: 1

When a lesson was completed, on the whole, how did you react to it?
I learned something new: 150
I didn't learn anything new but it was a good review: 60
I didn't learn anything new; it was a waste of time: 10
No response: 1
(Positive response is 95%)

Do you believe that doing the CAI lessons helped you to write better?
Yes: 55 Greatly: 5 A little: 116 No: 43 No Response: 1
(Positive response is 80%)

On the average, rate the difficulty of the questions:
Very Hard: 1 Hard: 136 Easy: 77
Very Easy: 2 Tricky: 2 Varied: 3

Did you ever do a lesson more than one time?
Yes: 120 No: 98 No Response: 3

If yes, was it because the instructor requested you to do so?
Yes: 27 No: 93

Would you as a student user recommend

the continued use of the CAI lessons in English:

Yes, on a required basis: 83
Yes, on an optional or referral basis: 126
No: 12
(Positive response is 94%)

The responses of the ten faculty members recommending the review modules coincides with those of the students:

Did you notice any improvement in the writing of students who performed the CAI lessons?
Always: 0 Sometimes: 8
Never: 1 No Response: 1

Would you as a teacher recommend the continued availability of the CAI English lessons?
Yes, on a required basis: 2
Yes, on an optional or referral basis: 8
No: 0

The programs have been retained on an optional or referral basis for students in "Composition and Literature," and their use has been found helpful in other freshman level courses where writing is an important component: the Humanities Seminar (10 sections) and the Freshman Seminar (52 sections). In some upper-division courses, the phonology modules are utilized in "Introduction to Linguistics," "Dialectology," and "History of the English Language," and the morphology-syntax modules are recommended to students in advanced expository prose writing classes. The entire system is utilized in "English as a Second Language" courses and is available to any student or faculty member who wishes to sharpen his or her grammatical skills.

The entire program is available to the public free of charge, except for the cost of one magnetic tape and processing changes, upon inquiry to the author.

FOOTNOTES

1. It is somewhat ironic to note that many so-called "grammar" schools have entirely abandoned the teaching of grammar in favor of grab bag of items called "communication skills," and many secondary schools have relegated even watered-down studies of grammar to a position where such courses must compete with so-called relevant elective minicourses such as "Science Fiction," "Comedy with Lucy and Desi," and "The Touchy-Feely School of Poetry Writing."

2. The author is indebted to the College of Arts and Letters, University of Notre Dame, for a grant in June, 1975 for the preparation of materials to use in Part I of the program. The author also acknowledges the development of the program format by Dr. William I. Davisson, Department of Economics, and valuable editorial assistance provided by Dr. Sandra Harmatiuk, Freshman Learning Resources Center, both of the University of Notre Dame.

APPENDIX I

Results of Student Evaluation
Fall Semester, 1976

NR indicates "no response." A category enclosed in parenthesis indicates one not present on the original questionnaire.

Total Number of Student Responses: 221

Part I. Using the Computer

1. Was the verbal explanation on use of the computer terminal clear?
   Yes: 213 No: 8
2. Was it difficult to LOGON the terminal?
   Yes: 46 No: 155
   (Sometimes: 13) NR: 7
3. Was it difficult LOGOFF the terminal?
   Yes: 20 No: 196 NR: 5
4. Was it difficult to access a particular review module?
   Yes: 15 No: 201 NR: 5
5. Did you encounter mechanical problems when using a terminal?
   Never: 37 Always: 16
   Sometimes: 166 (Mostly: 1)
   NR: 1
6. If you made a typing error, did you know what to do?
   Yes: 143 Sometimes: 41
   No: 33 NR: 4
7. Did you encounter delays in obtaining a terminal?
   Yes: 20 No: 101
   Sometimes: 99 NR: 1

Part II. The Lessons

1. Were the instructions on how to answer the questions clear?
   Yes: 211 No: 9
   (Sometimes: 1)
2. On the whole, the individual lessons were:
   Too long: 65 Too short: 0
   Average: 46 Varied: 110
3. On the whole, the instructional materials in the lessons were:
   Too long: 58 Too short: 4
   Average: 111 Varied: 47
   NR: 1
4. Did your instructor refer you to a handbook or grammar book before recommending that you do a particular CAI lesson?
   Always: 118 Never: 53
   Sometimes: 49 NR: 1

WHAT WILL HAPPEN IF ____?

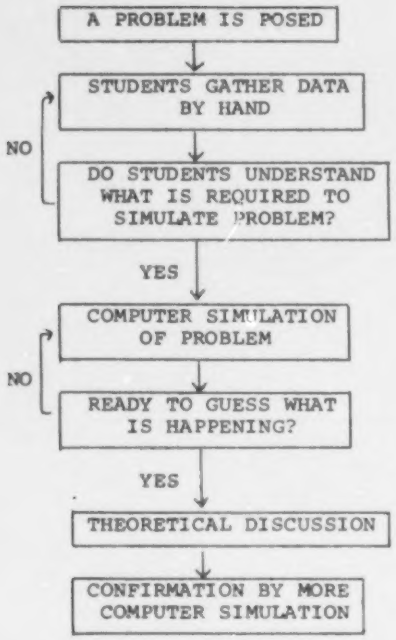
Glenda Lappan and M.J. Winter
Michigan State University

The paper will describe a method of teaching the concept of expected value through an experiment - conjecture - explain mode which uses the computer to simulate repetitions of games. The games, the rationale for choosing those particular games, and the computer programs to simulate the games will be given in detail.

WHAT WILL HAPPEN IF ____?

In the 1600's a French gambler, the Chevalier de Méré, had a run of "bad luck". The dice turned against him and he suddenly found himself in the uncomfortable position of losing money! For some time the Chevalier had been systematically winning by betting (at even odds) that in four tosses of a die he would obtain a six at least once. When his source of takers dried up, he changed the bet. He now bet that in 24 tosses of a pair of dice, he would obtain a double-six at least once. Since 4/6 = 24/36, he assumed he had an equally profitable bet. To his surprise he began to lose. In his search to find out why, he consulted Blaise Pascal (1623-1662) who in turn discussed the problem with Pierre de Fermat (1601-1665), two of the greatest mathematicians of all time. From the analysis of de Méré's problem a new branch of mathematics was born. Born of a desire to understand the behavior of systems that cannot be entirely controlled.

Just as the Chevalier started his study of probability with a misunderstanding about the probabilistic basis of his new bet, many students start their study of probability with misconceptions about the nature of a probabilistic statement. Their prior study of mathematics has dealt (at least in their minds) with exact answers. Students expect probability to deliver an exact description of what will happen. If a fair coin is tossed 10 times you will get exactly 5 heads. Consequently, (the students' reasoning goes) if you have tossed it 9 times and have gotten h,t,h,h,t,h,t,h,t, then of course the next toss has a high probability of being a t. If you ask a class on day one which outcome is more likely to happen (h,h,h,h,h,h,h,h,h,h) or (h,t,t,h,h,t,h,h,t,t), many of the students will choose the latter because 5 t's and 5 h's are more likely than 10 t's. True, but not relevant to the question asked. Each of the outcomes has probability $1/2^{10} = 1/1024$ of happening. Let us give another example from a game situation. If the probability of winning a game which pays even money is 3/7, the student expects to be exactly $1 behind at the end of 7 games. The student assumes that you win exactly 3 and lose 4. So you win $3 and lose $4, for a total loss of $1. Theoretically he is correct. It is in the belief that this theoretical expected value determines what will happen for each 7 games that the student is in error. The authors feel that students need to have their "faith" in what probabilistic statements mean shaken. To accomplish this, students need to do a great deal of experimentation before being exposed to a theoretical discussion of certain aspects of probability. That is, they need to explore a situation until they have some feeling for what is happening or until, as de Méré, they are puzzled enough to understand how the system works. The computer is an excellent tool for this type of experimentation. In the time it takes students to repeat an experiment 100 times, a computer may simulate the experiment 10,000 times. The thinking required to set up a computer simulation is in itself useful in understanding the experiment. A useful sequence of activities for students might look like this:


A PROBLEM IS POSED
STUDENTS GATHER DATA BY HAND
NO
DO STUDENTS UNDERSTAND WHAT IS REQUIRED TO SIMULATE PROBLEM?
YES
COMPUTER SIMULATION OF PROBLEM
NO
READY TO GUESS WHAT IS HAPPENING?
YES
THEORETICAL DISCUSSION
CONFIRMATION BY MORE COMPUTER SIMULATION

The rest of this paper describes a sequence of laboratory and hands-on computer activities designed to assist in the teaching and learning of one important aspect of the probability of game situations -- expected value. These activities are divided into two groups.

The first group comprises several games of "chance". The games, with names such as Gambler's Choice, Over and Under, Coins in a Pocket, have been programmed in BASIC, using the RND function. The outcomes depend on a roll of dice, a random sample, or a coin toss. Students play the games several times, as many as they think necessary in order to predict the result if each game were to be played 10,000 times. The games are designed so that for some, a prediction can easily be made, for others, not. A classroom discussion of expected value enables the students to verify the accuracy of some of their predictions. Extended discussion of the expected value of independent variables leads to successful analysis of the remaining games.

The second group develops intuitively the expected number of trials until first success, preparing the way for a classroom derivation of the formula 1/p. Simulation of completing a collection of baseball cards allows the students to compare a complicated example with the results predicted by theory.

ACTIVITIES

The unit is introduced by simulating **COINS IN A POCKET.** The story is that a newspaper costs 5¢. A customer has 5 pennies and a dime in his pocket and offers to pay for the paper by letting you, the vendor, select, at random, 2 of the 6 coins. Using marked chips, the students repeat the selection of coins 10 times and calculate the total value of the coins selected. One student might obtain 2 pennies 8 times and a dime and penny 2 times for a total of 38¢. Based on these 10 trials, the student would predict that after 100 trials she would have $3.80; in other words the average of 3.8¢ per trial would hold.

To actually repeat the game 100 times would be time-consuming, so we look for a way to do it by computer. On our machines, Tektronix 4051's, using BASIC, RND (-1) produces a random variable between 0 and 1, (the (-1) ensures that repetitions will produce different results). 6*RND(-1) lies between 0 and 6; so that X = INT(6*RND(-1)) produces a variable equally likely to take on the values 0,1,2,3,4,5. We identify selecting a coin with generating X; if X = 0 we'll say we selected the dime. The program we use is:

```
100 PRINT "TAKING TWO COINS AT RANDOM
    FROM A DIME AND 5 PENNIES"
110 PRINT
120 PRINT "HOW MANY TIMES SHALL WE
    REPEAT?"
130 INPUT N
140 U=0
150 FOR K=1 TO N
160 X=INT(6*RND(-1))
170 IF X=0 THEN 220
180 Y=INT(5*RND(-1))
190 IF Y=0 THEN 220
200 U=U+2
210 GO TO 240
220 U=U+11
240 NEXT K
250 PRINT "AVERAGE VALUE OF THE TWO COINS
    IS ";U/N;" CENTS"
260 END
```

Running this program for N = 100 leads most students to predict that after 1000 times the total amount would be close to 5000 cents. Coins in a Pocket is a good game situation to start with because the average value becomes apparent after a few runs of 100 trials. In the next sequence of games it is much harder to decide on the average value.

GAMBLER'S CHOICE: A gambler has a choice of two games. The first costs $10 to play; 3 dice are rolled and the player receives the sum of the numbers rolled in dollars. The second game costs $12 to play; 2 dice are rolled and the player receives the product of the numbers rolled in dollars. The program plays both games the same number of times.

```
100 PRINT "GAMBLERS CHOICE"
110 PRINT
120 PRINT "GAME 1 COSTS $10 TO PLAY.  3
    DICE ARE ROLLED."
130 PRINT "YOU GET THE NUMBER OF DOLLARS
    EQUAL TO THE SUM OF THE DICE"
140 PRINT
150 PRINT "GAME 2 COSTS $12 TO PLAY.  TWO
    DICE ARE ROLLED.  YOU GET THE"
160 PRINT "NUMBER OF DOLLARS EQUAL TO THE
    PRODUCT OF THE DICE"
170 PRINT
180 PRINT "HOW MANY TIMES DO YOU WANT TO
    PLAY EACH GAME?"
190 INPUT N
200 I=0
210 J=0
220 FOR K=1 TO N
230 X=INT(6*RND(-1))+1
240 Y=INT(6*RND(-1))+1
250 Z=INT(6*RND(-1))+1
260 I=I+X+Y+Z-10
270 J=J+X*Y-12
280 NEXT K
290 PRINT "IN GAME 1 YOUR WINNINGS TOTAL
    ";I;"AVERAGE PER GAME=";I/N
300 PRINT
310 PRINT "IN GAME 2 YOUR WINNINGS TOTAL
    ";J;"AVERAGE PER GAME=";J/N
320 END
```

Typical outputs are:

AVERAGE WINNINGS PER GAME

N	Game 1	Game 2
1000	.58	.41
1000	.45	.20
5000	.54	.35

For the game one the students might say, "There appears to be an average value and it appears to be in the low 50's." For game two the data presents a much more confusing picture.

OVER AND UNDER: This game has been popular at fundraising events. Two dice are rolled; the player can bet that the sum of the numbers showing will be under 7, equal to 7, or over 7. Bets on over and under each pay even money; 7 pays four times the bet. If you always bet over (or under) what will your winnings be after 100 games? After 10,000 games? Will you do better if you always bet on 7? (Losing less *is*, mathematically, doing better!) For a series of 100 games, the outcome is clear, but the amount of the average loss is not clear.

```
100 DIM W(3)
110 PRINT "OVER AND UNDER"
120 PRINT "HOW MANY GAMES DO YOU WANT TO
    PLAY?"
130 INPUT N
140 PRINT "HOW DO YOU BET? FOR UNDER,
    ENTER -1; FOR 7 EXACTLY, ENTER 0;"
150 PRINT "FOR OVER, ENTER 1"
160 INPUT B
170 W(1)=0
171 W(2)=0
173 W(3)=0
180 FOR J=1 TO N
190 C=INT(6*RND(-1))+1
200 D=INT(6*RND(-1))+1
210 IF C+D<7 THEN 250
220 IF C+D=7 THEN 270
230 W(3)=W(3)+1
240 GO TO 300
250 W(1)=W(1)+1
260 GO TO 300
270 W(2)=W(2)+1
300 NEXT J
310 IF B=-1 THEN 360
320 IF B=0 THEN 390
330 G=W(3)-W(2)-W(1)
340 GO TO 430
360 G=W(1)-W(2)-W(3)
370 GO TO 430
390 G=4*W(2)-W(1)-W(3)
430 PRINT "YOUR TOTAL WINNINGS ARE $";G
440 PRINT "AVERAGE WINNINGS PER GAME ARE
    $";G/N
450 END
```

BETTING ON 7		BETTING ON OVER	
Number of Games	Average Winnings	Number of Games	Average Winnings
100	-.25	100	-.22
100	-.35	100	-.20
100	-.20	100	-.08
100	0	100	-.12
100	-.5	100	-.20

After the students have played these games and have tried to predict the outcome for 10,000 or 100,000 trials, we emphasize that their predictions are made on the basis of average winnings and then consider the following situation: Let 2 coins be tossed. If they both show heads, you will win \$5. If one is heads and one tails, you will win \$2, but if they both show tails, you will lose \$20. Suppose the game has been played 1000 times with the outcomes:

Heads, Heads 200 times
Heads, Tails 721 times
Tails, Tails 79 times

Your winnings will be

$$\text{winnings} = 5\cdot 200 + 2\cdot 721 + (-20)\cdot 79$$

Your average winnings per game will be

$$\text{average winnings} = \frac{\text{winnings}}{1000} =$$

$$5\cdot\left[\frac{200}{1000}\right] + 2\cdot\left[\frac{721}{100}\right] + (-20)\left[\frac{79}{1000}\right] = .862.$$

The bracketed terms, $[\frac{200}{1000}]$, $[\frac{721}{1000}]$ and $[\frac{79}{1000}]$ are the relative frequences of 2 Heads, Heads and Tails, and 2 Tails, respectively. Over a great number of trials, these relative frequencies will approach the theoretical probabilities of these events. When the relative frequencies are replaced by probabilities, we call the average winnings, the expected value of the game.*

Expected Value = E = payoff x probability + payoff x probability +...

Now we will look at the games played and compute theoretical expected values. For coins in a pocket, the probability of 2 pennies is $\frac{10}{15}$, the probability of a dime and penny is $\frac{5}{15}$. Thus the expected value is $2 \times \frac{10}{15} + 11 \times \frac{5}{15} = \frac{75}{15} = 5$ cents. This confirms the student predictions based on the simulations.

For Over and Under, the probability of a 7 is $\frac{6}{36} = \frac{1}{6}$; the probability of over is $\frac{15}{36}$; the probability of under is also $\frac{15}{36}$. For the games, assuming a \$1 bet,

$$E \text{ (bet on 7)} = 4 \times \frac{1}{6} + (-1) \times \frac{5}{6} = -\frac{1}{6}$$
$$= -.1666$$
$$E \text{ (bet on over)} = 1 \times (\frac{15}{36}) + (-1) \times (\frac{21}{36})$$
$$= -\frac{1}{6} = -.1666$$

The student predictions based on the simulations cited earlier might be -.26 for a bet on 7 and -.16 for a bet on under. After the theoretical discussion, the students might decide to simulate the experiment again using the computer for a larger number of trials.

For Gambler's Choice I and II, the possible outcomes can be listed and their relative frequencies (i.e. theoretical probabilities) can be determined. For example in Game 1, we get the following theoretical frequencies

Sum	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Frequency	1	3	6	10	15	21	25	27	27	25	21	15	10	6	3	1

$$E(\text{game } 1) = (-7) \times (\frac{1}{216}) + (-6) \times (\frac{3}{216}) + (-5) \times (\frac{6}{216}) + (-4) \times (\frac{10}{216}) + (-3) \times (\frac{15}{216}) \ldots \text{ etc.}$$

$E(\text{game } 1) = .50$.

These calculations are tedious and serve as a motivation for the theoretical analysis of expected value of independent events. If X, Y and Z are the values showing on a die, then $E(X + Y + Z) = E(Y) + E(Z)$ and $E(XY) = E(X)E(Y)$. The expected value on the face of a die is $\frac{1+2+3+4+5+6}{6} = 3.5$. For game 1, subtracting the \$10 to play, the expected value is $3.5 + 3.5 + 3.5 - 10 = .5$ or \$.50. For game 2 it is $(3.5) \times (3.5) - 12 = 12.25 - 12 = .25$. From the simulations, the former value was predicted by many students; the .25 did not become apparent even after 10,000 trials.

TRIALS UNTIL FIRST SUCCESS

One of the first applications of expected value is determining the mean number of trials until first success. The students first use the computer to simulate several examples. The program used is:

```
LIST
100 PRINT "NUMBER OF TRIALS UNTIL FIRST
     SUCCESS"
110 PRINT "ENTER P, THE PROBABILITY OF
     SUCCESS, AS A DECIMAL"
120 INPUT P
130 PRINT "ENTER N, THE NUMBER OF
     EXPERIMENTS"
140 INPUT N
150 T=0
160 FOR J=1 TO N
170 K=0
180 X=RND(-1)
190 K=K+1
200 IF X<P THEN 220
210 GO TO 180
220 T=T+K
230 NEXT J
240 PRINT "AVERAGE NUMBER OF TRIALS UNTIL
     SUCCESS IS ";T/N
250 END
```

Then if p is the probability of success and $q = 1 - p$ the probability of failure, the expected number of trials is

$$E = 1 \cdot p + 2q \cdot p + 3q^2 \cdot p + 4q^3 \cdot p + \ldots,$$

where $1 \cdot p$ stands for 1 trial x probability of success, $2 \cdot q \cdot p$ stands for 2 trials x probability of 1 failure then success, $3 \cdot q^2 \cdot p$ stands for 3 trials x probability of 2 failures then success. A standard manipulation, computing $E - qE$, summing the geometric series and solving for E, leads to

$$E = \frac{1}{1-q} = \frac{1}{p}$$

If the probability of success is $\frac{1}{6}$, e.g. as in rolling a 4, then it will take, on the average $\frac{1}{p} = 6$ tries until the first success, i.e. until the first 4. An application of this result is the number of baseball cards needed before acquiring a complete set. We assume there is an unlimited supply of N different baseball cards. The expected number of cards necessary to have a complete set is

$$\frac{N}{N} + \frac{N}{N-1} + \frac{N}{N-2} + \ldots + \frac{N}{1} = N(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{N}).$$

For $N = 10$, $E = 29.29$. For $N = 50$, $E = 224.96$. The program listing is

```
100 DIM C(100)
110 PRINT "BASEBALL CARDS. THERE ARE N
    CARDS IN THE SERIES. ENTER N"
120 INPUT N
130 PRINT "FROM HOW MANY TRIALS DO YOU
    WANT TO COMPUTE THE AVERAGE?"
140 INPUT K
150 S=0
160 FOR I=1 TO K
170 D=0
180 FOR J=1 TO N
190 C(J)=0
200 NEXT J
210 X=INT(N*RND(-1))+1
220 C(X)=C(X)+1
230 IF C(X)=1 THEN 250
240 GO TO 210
250 D=D+1
260 IF D=N THEN 280
270 GO TO 210
280 T=0
290 FOR J=1 TO N
300 T=T+C(J)
310 NEXT J
320 PRINT T;" CARDS"
330 S=S+T
340 NEXT I
350 PRINT "AVERAGE NUMBER OF CARDS IS
    ";S/K
360 END
```

Sample output from the BASEBALL CARDS program is:

```
BASEBALL CARDS. THERE ARE N CARDS IN THE
SERIES. ENTER N
10
FROM HOW MANY TRIALS DO YOU WANT TO
COMPUTE THE AVERAGE?
15
 29 CARDS
 31 CARDS
 34 CARDS
 17 CARDS
 64 CARDS
 44 CARDS
 20 CARDS
 30 CARDS
 23 CARDS
 38 CARDS
 30 CARDS
 16 CARDS
 31 CARDS
 23 CARDS
 28 CARDS
 AVERAGE NUMBER OF CARDS IS 30.5333333333
```

Let us return to the Chevalier de Méré's problem and compute the expected winnings of a bet of a louis d'or in each situation.

The probability of at least one six in four rolls of a fair die is $1 - (\frac{5}{6})^4 = .5177$, so that the expected winnings are

$$E = 1 \times .5177 + (-1) \times .4823 = .0354.$$

But, the probability of a double six in twenty four rolls of a pair of fair dice is $1 - (\frac{35}{36})^{24} = .4914$, so that the expected winnings are

$$E = 1 \times .4194 + (-1) \times .5086 = -.0172$$

Computer simulations make one wonder how much time the Chevalier spent gambling since the difference in the two bets is not perceivable in 10,000 trials!

*For this example,

$$E = 5 \times (\tfrac{1}{4}) + 2 \times (\tfrac{1}{2}) + (-20) \times (\tfrac{1}{4})$$

$$= -\frac{11}{4} = -3.75$$

In the (very) long run you will average a loss of \$2.75 per game.

LOW COST CAI WITH A MICROCOMPUTER

by

Daniel J. Macero and Francis C. McElroy

Department of Chemistry
Syracuse University, Syracuse, N.Y. 13210

Little attention has been directed toward the use of a microcomputer system as a vehicle for carrying out computer aided instruction. We have designed and implemented such a system. The microcomputer was assembled from components purchased in kit form and features an Z80 microprocessor chip, 26 kilobytes of RAM, a 256-byte PROM bootstrap loader, a TV readout and audio cassette interface, and 8 I/O ports. Inexpensive large-scale data storage and retrieval is accomplished with computer-controlled cassettes which can store up to 500,000 8-bit bytes on a single C-60 cassette. All programming is done in BASIC. The hardware and software problems inherent in getting started with such a system are presented along with examples of several computer assisted instructional programs implemented on the microcomputer system.

Introduction

A revolution in digital electronics has occurred over the past decade. Advances in medium scale (MSI) and large scale integrated (LSI) circuit technology have resulted in the development of low cost individual chips capable of carrying out operations that formerly required many hundreds, even thousands, of discrete electronic components [1].

The most remarkable of the new LSI devices is the single chip microprocessor, which is an integrated circuit that contains at least 75 percent of the power of a very small computer [2]. The addition of memory,control circuitry, data input and output lines to a microprocessor results in a microcomputer, i.e., a fully operational computer system.

Already several reports dealing with the application of microprocessors to chemical instrumentation have appeared. Woodward, Ridgway, and Reilley [3] describe a microprocessor-based data acquisition system for the automated laboratory where the microcomputer serves primarily as an "intelligent" terminal. Betteridge et al. [4] designed an automatic potentiometric titrator controlled by a microprocessor, which has been used successfully in the determination of mixtures of adipic and boric acids. Similarly a microprocessor-controlled coulometric titrator was developed by Earle and Fletcher [5] for process control applications. Pardue et al. [6] employ a microcomputer slaved to a minicomputer in a hierarchical computer controlled stopped-flow spectrophotometer. More recently, Malmstadt and coworkers, describe the use of a microprocessor to control the wavelength selection of a scanning dye laser [7]. Macero and McElroy [8-10] have developed a microcomputer controlled voltammetry system which features experimental parameter control, data storage and retrieval, data averaging, and various data plotting and data reduction routines.

At present most applications of microcomputers to chemical education deal with their use in teaching computer hardware principles, instrument interfacing and laboratory automation [11,12]. Little attention has been directed toward the use of a microcomputer system as a vehicle for carrying out computer aided instruction. This is not surprising since the microprocessor industry has undergone a period of very rapid growth with new microprocessor and microcomputer systems being announced daily. Under these conditions the choice of hardware becomes confusing, not only because of the variety of items available, but also, because of the fear that equipment once purchased may soon become obsolete.

Along with this expanding use of microprocessors in scientific applications is the rapid development of microcomputer systems primarily aimed at providing low cost computing. These efforts have been so successful that it is now possible to purchase for under $2500, a computer system which features a keyboard with video readout, an audio cassette recorder interface for loading and storing programs, and up to 32 kilobytes of random access memory.

These developments in computer technology prompted us to consider assembling such a system and to explore its use for computer aided instruction. Our aim was to assemble a versatile, low cost computer system independent of the campus computing center, and yet not sacrifice any of the features usually associated with larger, more expensive, computer installations, i.e., video and hardcopy alphanumeric and graphical input/output, data files, and interactive student-computer operation. To achieve this goal it was necessary to configure a stand alone microcomputer; one with sufficient memory, rapid large-scale storage and retrieval capability, and a high level, interactive programming language.

Hardware

In late 1975 when we made our decision to select a microcomputer system for CAI use, the major chips available were the Intel 8080, the Mostek 6502 and the Motorola 6800, all 8-bit devices. We elected to use a system featuring the 8080 chip because, at the time, this processor gave every indication of having better software support, was being second-sourced by many manufacturers and appeared to be taking over as the leader in the field. As it happened this choice was especially fortunate, for, when we changed over to the faster and more powerful Zilog Z80 microprocessor in early 1977, we were able to use much of our existing 8080 software since the Zilog chip is software compatible with the 8080. An 8-bit processor was selected in favor of a 16-bit device to take advantage of the lower hardware cost of the unit with the smaller word size.

A block diagram of the microcomputer-based CAI system is shown in Figure 1. The microcomputer was purchased in kit form from the Digital Group, Inc., Denver, Colorado 80206. The CPU board (Digital Group No. Z80-CPU) consisted of a Z80 microprocessor, a 256-byte erasable read only memory bootstrap loader (EROM), 2 kilobytes of static random access memory (500 ns 2102's), and miscellaneous drivers, decoders and gates. The CPU runs at 2.5 MHz thus yielding a machine cycle time of 400 nsec. Included on the CPU board is the necessary hardware for priority interrupts and full direct memory access capability, along with that required to implement such functions as POWER ON RESET and SINGLE STEP. Buffering is provided for driving up to 65 K bytes of memory and up to a maximum of 256 input/output ports.

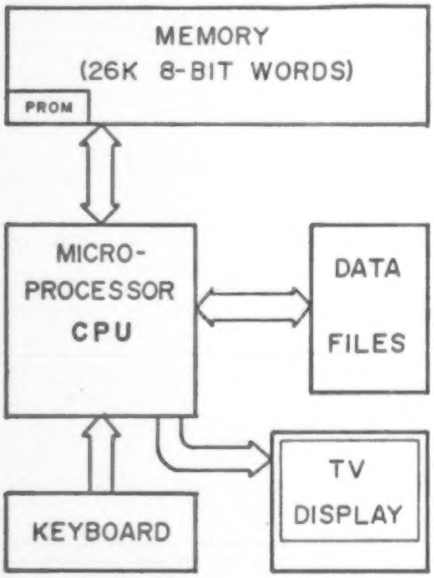


Figure 1
MICROCOMPUTER SYSTEM
FOR CAI

The TV-cassette interface (DG No. TVC-64) consists of two sections: the first section is used to output data from the computer to a video monitor (Sanyo Model VM 4209) or modified TV set [13], the second interfaces to an audio cassette recorder (Superscope C-104) and allows programs to be saved on and loaded from standard Phillips type magnetic cassettes [14]. The TV readout has a 128 character ASCII set, generates 64 characters (each in a 7 x 9 dot matrix) horizontal by 16 lines, has a software driven cursor and 1 K of on-board RAM for buffer storage. The cassette interface uses FSK recording and runs at 1100 baud or 100 characters per second. An ASCII encoded capacitance keyboard (DG No. KEY1-COMP) with numeric pad and cursor control keys is used for manual program and data entry.

Twenty-six K bytes of random access memory are obtained from three 8K memory boards (DG No. MEM-8C), each containing sixty-four 2102 IC's, and from 2 K bytes of memory located on the CPU board. Available for interfacing various peripheral devices to the computer are eight I/O ports which are derived from two input/output boards (DG No. IO-F), each consisting of four latchable, 8-bit input ports and four 8-bit output ports.

A computer-controlled digital cassette storage system (DG No. PHI2-COMP) was also interfaced to the microcomputer. The interface for this system can control up to 4 cassette drives (PHI-DECKS) and allows for the storage of approximately 500,000 8-bit bytes on a single C-60 cassette with a soft error rate of less than one bit in 10^8. The search speed is 100 inches per second for a worst case search time of approximately 20 seconds; the tape speed is 5 inches per second for a data recording and reading rate of 800 bytes per second (6400 BAUD.)

Software

Advances in microcomputer software have not kept pace with those in hardware. This problem has been alleviated by the appearance of high level languages such as BASIC and FORTRAN which are now available for use with many microcomputer systems.

We have in hand a version of extended BASIC (Digital Group MAXI-BASIC Version 1.1) which we are using with our Z80-based microcomputer. BASIC is a widely used, algebraically oriented, easy-to-learn programming language. It is conversational and interactive so that programs can be entered, executed, and edited on-line rapidly and easily. Another advantage of using BASIC is that it is a common programming language for CAI with the result that existing CAI programs [15] written in BASIC can be easily adapted for use on our system. Likewise new BASIC CAI programs developed on our system, can be readily adapted to other hardware configurations whether they are micro-, mini- or maxicomputer systems.

MAXI-BASIC and the operating system reside in the first 63 (octal) pages of memory, i.e., approximately 13 K bytes of RAM. In operation MAXI-BASIC automatically searches for the top of memory and adjusts itself for any size of continuous memory.

MAXI-BASIC is an extended BASIC and is able to handle such features as multiple-dimensional arrays, strings, formatted output, and machine language calls. It is line-number oriented with the ability to include multiple program statements on a single line if these are separated by a colon. It has most of the usual BASIC commands, i.e., RUN, LIST, SCR (scratch or delete the entire program), REN (renumber the existing program), CLEAR (clear all variables), CONT (continue), LINE (line length), SAVE (save a program on to a cassette), LOAD (load a program from a cassette. MAXI-BASIC statements include: LET; IF, THEN, ELSE; FOR, NEXT; GOTO; ON; REM; READ, DATA; RESTORE; INPUT; GOSUB, RETURN; END and PRINT. Variables or expressions to be printed, when separated by a comma, output five spaces between variables and no spaces when separated by a semicolon. Formatting may be accomplished by including a format string in a print statement.

Arrays may be dimensioned with any number of dimensions and are limited only by the size of memory. Strings of 8-bit characters also may be dimensioned to any size within the constraint of available memory. Strings, substrings and string expressions may be used in conjunction with the common BASIC statements: LET, READ, DATA, PRINT, IF, and INPUT. The string IF does alphabetic comparisons when the relational operators, <,>,<>,=, are used.

Numeric operators available are: addition, (+), subtraction, (-), multiplication, (*), division, (/), and exponentiation, (**) or (↑). The Boolean operators, AND, OR, and NOT, return as 1 if true or 0 if false. Several built in functions, e.g., ABS, INT, SIN, COS, RND, LOG, EXP, SQRT, etc. are available for use in addition to user defined functions which may be single line or multiple line functions.

MAXI-BASIC has two features which make it highly desirable for use in writing CAI programs. The first is that programs can be halted during execution by inserting a STOP statement at any point in the program, and the second is the ability to immediately execute typed statements when in command mode. The sign (#) may be used as a shorthand way of typing the PRINT reserved word. Accordingly when MAXI-BASIC is in command mode, the system may be used to carry out and display calculations by directly executing a PRINT statement which includes the designated mathematical operation. For example, by typing:

```
#5*3/2  RETURN
```

the answer: 7.5 is displayed on the next line. By typing CONT the program is resumed starting with the line immediately following the STOP statement.

Recently, we acquired a digital cassette-based operating system, PHIMON [16], for use with our Z80-based microcomputer and PHI-DECKS. PHIMON provides a set of executive routines which allow easy access to files and ready file creation. In operation, when the POWER ON switch is activated, PHIMON loads itself onto the system (with the aid of a 256-byte PROM bootstrap loader residing at memory locations, 0-256), self-configures to utilize all available memory, and displays the prompting character, >, when ready. A set of keyboard monitor commands, e.g., HELP, DIRECTORY, LOAD, RUN, START, etc., are available for use. A listing of these commands may be displayed by typing HELP and RETURN.

When running CAI programs, a master digital tape (SYNCOM SSX 300), on which PHIMON, MAXI-BASIC and various CAI programs have been stored, is inserted into the PHI-DECK and the computer turned on. After the system loads itself into memory and the PHIMON prompt character appears on the screen, a listing of the CAI program library may be displaced by entering DIRECTORY and RETURN. Such a listing is displayed in Figure 2 below.

```
PIP      .GO    7
MAXBAS   .GO   63
MOLES    .BA   20
SYMM     .BA   50
GROUPS   .BA   40
ACIDBA   .BA   32
SOLPRO   .BA   27
MCHOIC   .BA   40
```

Figure 2. Display of CAI Directory

PIP is the PHIMON system file utility program that permits copying programs from one device to another and allows the creation of new files from input. Files may be named with up to six alphanumeric characters, starting with an alphabetic character, followed by a period and an extension of two alphanumeric characters. The extension .GO refers to a memory run file, and .BA to saved BASIC programs. The two digit number to the right of the file name refers to the number (octal) of pages in memory the program occupies.

Discussion

For our initial venture into microcomputer assisted instruction we devised a tutorial program on qualitative analysis, since, at that time, our only high-level programming language was TINY BASIC [16] which could execute strings and relational operators, but could not handle floating point arithmetic. The program only covers Group

I of the standard qualitative analysis scheme. A printout of the initial portion of the student computer interaction for the qualitative analysis program is shown in Figure 3.

```
QUALITATIVE CHEMICAL ANALYSIS
   OF AG+, PB+2, AND HG2+2
THE AVAILABLE REAGENTS ARE:

1)  HOT WATER    5)  HCL 6M
2)  COLD WATER   6)  HNO3 CONC
3)  CRO4=        7)  RESTART
4)  NH3 AQ       8)  STOP

ADD REAGENT ? 5
--------RESULTS OF TESTS--------

WHITE PPT FORMS WITH 6M HCL
ADD REAGENT ?_
```

Figure 3. Portion of CAI Qualitative Analysis Program on Video Screen

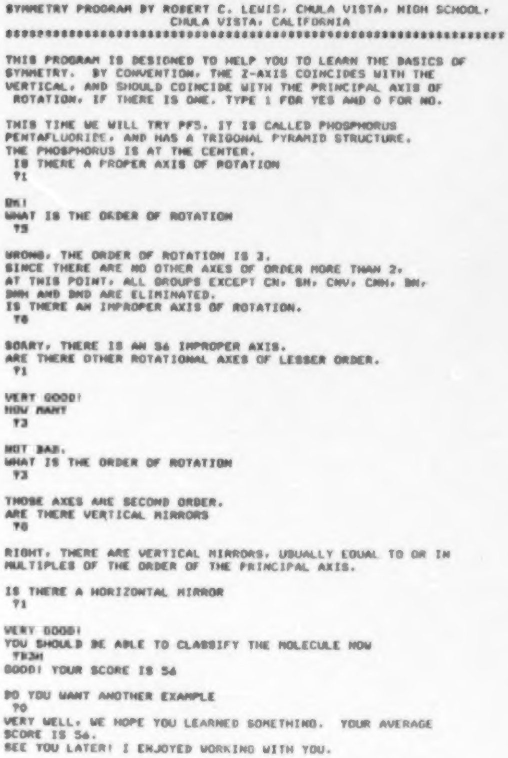
Note that student answers are input by selecting a number which corresponds to the "reagent" which he wishes to add to the system. As the student-computer session proceeds the display scrolls up and the topmost line eventually disappears from the screen when the number of lines exceeds 16. Note also the commands: RESTART and STOP which permit the student to review the example in question or to terminate the session, respectively. Figure 4 shows the student selecting his final answer and ending the session.

```
TO THE 1ST PPT
 SOME PPT DISS IN HOT WATER
TO THE REMAINING SOLUTION
 YELLOW PPT FORMS WITH CRO4=
TO THE 1ST PPT
 NR WITH AQ NH3
TO THE REMAINING SOLUTION
 WHITE PPT FORMS WITH HNO3
THESE ARE THE POSSIBLE UNKNOWNS

1)  AG+            5)  AG+,PB+2
2)  HG2+2          6)  HG2+2,PB+2
3)  PB+2           7)  AG+,HG2+2,PB+2
4)  AG+, HG2+2

WHAT IS YOUR CHOICE?  5
RIGHT!!!
LIKE TO TRY AGAIN (Y=1,N=0) ?
```

Figure 4. Qualitative Analysis CAI Program

In Figure 5 is shown a portion of a student-compuer session of a CAI program on molecular symmetry developed for a senior level course in quantum chemistry. Initially, this program was written in Digital Equipment Corporation System 10 BASIC and implemented on Syracuse University's DEC 10 maxicomputer. The symmetry program was later typed into our Z80 microcomputer system and operated satisfactorily after requiring only minor changes in the program.

Our principal goal in this project was to ascertain the feasibility of using microcomputers for computer aided instructional purposes, therefore, we sought to adapt existing, well-tried



```
SYMMETRY PROGRAM BY ROBERT C. LEWIS, CHULA VISTA, HIGH SCHOOL,
                    CHULA VISTA, CALIFORNIA
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

THIS PROGRAM IS DESIGNED TO HELP YOU TO LEARN THE BASICS OF
SYMMETRY.  BY CONVENTION, THE Z-AXIS COINCIDES WITH THE
VERTICAL, AND SHOULD COINCIDE WITH THE PRINCIPAL AXIS OF
 ROTATION, IF THERE IS ONE. TYPE 1 FOR YES AND 0 FOR NO.

THIS TIME WE WILL TRY PF5. IT IS CALLED PHOSPHORUS
PENTAFLUORIDE, AND HAS A TRIGONAL PYRAMID STRUCTURE.
THE PHOSPHORUS IS AT THE CENTER.
 IS THERE A PROPER AXIS OF ROTATION
 ?1

OK!
WHAT IS THE ORDER OF ROTATION
 ?5

WRONG, THE ORDER OF ROTATION IS 3.
SINCE THERE ARE NO OTHER AXES OF ORDER MORE THAN 2,
AT THIS POINT, ALL GROUPS EXCEPT CN, SN, CNV, CNH, DN,
DNH AND DND ARE ELIMINATED.
IS THERE AN IMPROPER AXIS OF ROTATION.
 ?0

SORRY, THERE IS AN S6 IMPROPER AXIS.
ARE THERE OTHER ROTATIONAL AXES OF LESSER ORDER.
 ?1

VERY GOOD!
HOW MANY
 ?3

NOT BAD.
WHAT IS THE ORDER OF ROTATION
 ?2

THOSE AXES ARE SECOND ORDER.
ARE THERE VERTICAL MIRRORS
 ?0

RIGHT, THERE ARE VERTICAL MIRRORS, USUALLY EQUAL TO OR IN
MULTIPLES OF THE ORDER OF THE PRINCIPAL AXIS.

IS THERE A HORIZONTAL MIRROR
 ?1

VERY GOOD!
YOU SHOULD BE ABLE TO CLASSIFY THE MOLECULE NOW
 ?D3H
GOOD! YOUR SCORE IS 56

DO YOU WANT ANOTHER EXAMPLE
 ?0
VERY WELL, WE HOPE YOU LEARNED SOMETHING.  YOUR AVERAGE
SCORE IS 56.
SEE YOU LATER! I ENJOYED WORKING WITH YOU.
```

Figure 5. Sample Printout of the CAI Program on Molecular Symmetry.

BASIC programs to our system, rather than develop new ones. An excellent collection of such BASIC programs is found in the book, CHEMISTRY WITH A COMPUTER BY Paul A. Cauchon [15]. Accordingly, several representative programs from this book were entered into our microcomputer, debugged, and stored on digital cassettes for later use.

The first program tried was a multiple choice type drill program. It is presented below as Figure 6.

```
YOUR NAME, PLEASE?D.  MENDELEEV
(1)
WHICH ELEMENT HAS CHEMICAL PROPE
RTIES MOST SIMILAR TO O?

TE     F     N     SB
YOUR ANSWER? TE
RIGHT?
(2)
WHICH ELEMENT HAS THE LARGEST AT
OMIC RADIUS?
SB     N     F     I
YOUR ANSWER? _
```

Figure 6. Example of a Multiple Choice Type CAI Program [15]

This program was run using a video display with 32 characters per line. Consequently, those lines which contain more than this number of characters are truncated at 32 and continued on the next line. This might be expected to be disconcerting to users; however, no adverse affect on thier performance could be detected. We have recently converted to a 64-character per line format. (See Figure 5).

In Figure 7 and 8 are shown two methods of carrying out calculations when engaged in a CAI program. The first is to use a pre-programmed approach [15] in which the student is asked if he or she needs math help. If the answer is yes, the program further inquires if a multiplication or division operation is desired. In Figure 7 division was requested in which case the computer responds with the prompts, DIVISOR= and DIVIDEND=. As might be expected users have trouble deciding which is the divisor and which is the dividend. In addition, this method of making calculations is time-consuming and somewhat cumbersome.

```
HOW MANY MOLES IN 1.15374E+25
MOLECULES OF A SUBSTANCE?
NEED MATH HELP? (Y OR N)?Y
MULTIPLICATION OR DIVISION?
(M OR D)? D
DIVISOR= 6.023E23
DIVIDEND = 1.154E25
 1.154E+25/ 6.023E+23= 19.15989
MOREHELP? (Y OR N)?N
WHAT IS YOUR ANSWER?  19.16
CORRECT!
WANT ANOTHER PROBLEM? (Y OR N)?_
```

Figure 7. Programmed Method for Carrying Out Calculations [15]

A better alternative is to have the students use an electronic hand calculator when attempting CAI programs requiring computations. A third alternative, made possible by the ability of MAXI-BASIC to directly execute individual BASIC statements, is to insert a STOP in the program where a numerical calculation is required and allow the student to use the direct execution option. This procedure is shown in Figure 8.

```
HOW MANY MOLES IN 1.97959E+24
MOLECULES OF A SUBSTANCE?
STOP IN LINE 2505
#1.98E24/6,023E23
 3.28740
READY
CONT
WHAT IS YOUR ANSWER?  3.287
CORRECT!
```

Figure 8. Calculation of Answer by Executing a Print Statement

Finally, an example of a drill program which has the capability of responding to incorrect answers and giving the student assistances in understanding the concept(s) involved is shown in Figure 9 where an interactive computer-student interchange takes place.

```
(1)
WHAT IS THE PH OF A .0065M SOLUT
ION OF HCL? 6.5
NO -
THE [H+] IS 6.5 x 10E-3, SO WHAT
IS THE LOG OF 6.5?  0.8129
RIGHT!  NOW SUBTRACT THIS FROM 3
AND YOU GET THE PH, 2.188.  STUDY
THAT, THEN HIT 'RETURN'?
(2)
WHAT IS THE PH OF .0024M HYDROCH
LORIC ACID?2.62
RIGHT!
```

Figure 10. CAI Tutorial Type Program [15]

Summary

In summary, our experience in using a stand alone microcomputer system for computer aided instructional purposes has been very favorable. All indications are that such systems will become common in the future and will find wide use as vehicles for CAI because of their low-cost, versatility, and reliability. An additional feature certain to accelerate this trend is that these systems can be easily adapted to other uses when not employed in CAI, e.g., instrument interfacing, experiment control, and data taking.

We anticipate also that the addition of peripheral devices such as graphics terminals, graphics, plotters, printers and floppy disks, to microcomputer CAI systems, will become standard despite the fact that such systems can no longer be termed "low cost", i.e., under $3000; however, the advantages accruing from such additions will be well worth the extra expense. For this reason, we interfaced a Tektronix 4013 alphanumeric and graphics CRT terminal to our microcomputer. For hard copy alphanumeric and graphical output we also incorporated into our system a Digital Equipment Corporation LA36 DECwriter II terminal and a Hewlett-Packard Model 7001A X-Y recorder, respectively. The CRT and printing terminals require a universal synchronous receiver/transmitter interface while the recorder is serviced by a 12-bit digital-to-analog converter. These additional devices considerably enhance the teaching and learning options available for carrying out high-quality computer assisted instruction.

Acknowledgement

We wish to thank Robert C. Lewis for providing us with the CAI programs on Groups and Symmetry and Lloyd W. Burgess, Jr. and Thomas M. Banks for their invaluable assistance in getting the digital cassette storage and retrieval system software operational. Our thanks also goes to Dudley Chapman for much advice and consultation concerning miscellaneous hardware and software problems during the course of this work. Finally, we are grateful to the Syracuse University Graduate School and the Department of Chemistry for providing some of the funds used to implement this project.

References

1. R. N. Noyce, SCIENCE, 195, 1102 (1977).
2. D. G. Larsen, P. Rony, and J. Titus, The Bugbook III: Microcomputer Interfacing Experiments Using the Mark 80 and 8080 System, E & L Instruments, Inc., Derby, Conn., 1975, p. 1-13.
3. W. S. Woodward, T. H. Ridgway, and C. N. Reilley, Analyst, 99, 838 (1974).
4. D. Betteridge, E. L. Dagless, P. David, D. R. Deans, G. E. Penketh, and P. Shawcross, Analyst, 101, 409 (1976).
5. W. E. Earle and K. S. Fletcher III, Chemical Instrumentation, 7(2), 101 (1976).
6. G. E. Mieling, R. W. Taylor, L. G. Hargis, J. English, and H. L. Pardue, Anal. Chem., 48, 1686 (1976).
7. J. A. Perry, M. [illegible] Bryant, and H. V. Malmstadt, Anal. Chem., 49, 1702 (1977).
8. F. C. McElroy and D. J. Macero, Paper No. 138, Division of Analytical Chemistry, 173rd National Meeting, American Chemical Society, New Orleans, La., April, 1977; Paper No. 9, Div. of Computers in Chemistry, 174th National Meeting, ACS, Chicago, Illinois, August, 1977.
9. D. J. Macero and F. C. McElroy, Abstracts, American Chemical Society Workshop on Computers in Chemistry III, Montclair State College, Montclair, New Jersey, June, 1977.
10. D. J. Macero and F. C. McElroy, Abstracts, Eastern Analytical Symposium, New York, N. Y., December, 1977.
11. R. E. Dessy, SCIENCE, Vol. 192, 511 (1976).
12. J. T. Arnold, SCIENCE, Vol. 192, 519 (1976).
13. R. Suding, BYTE, #12, 66, (1976).
14. R. Suding, BYTE, #11, 46, (1976).
15. P. A. Cauchon, Chemistry With a Computer, Educomp, Corporation, Hartford, Connecticut (1976).
16. PHIMON Operating System Manual, D. Bryant, The Digital Group, Inc. (1977).
17. R. Whipple and J. Arnold, People's Computer Company, 4(2), (1975).

COMPUTERIZING READING: A REVIEW

George E. Mason
Jay S. Blanchard
University of Georgia

In this paper is presented a year-by-year review of the highlights of fourteen years of efforts to use computers to aid in the improvement of the reading skills of children and adults. The strengths and weaknesses of computer-based reading instruction are assessed and future applications predicted.

The first well-publicized effort to use computers for reading instruction was the project begun at Stanford's Institute for Mathematical Studies in the Social Sciences in 1964. Atkinson, Fletcher, Suppes, and their colleagues and students first designed a tutorial program for beginning reading instruction that involved cathode ray tube presentations supplemented by computer-controlled audio.[1, 2] The students responded by pointing light pencils at the screen. However, the cost of the program was high and the Stanford group soon turned to supplement the program and materials used by teachers in the primary grades.[3]

Only two years later, in 1966, the first use of the computer in a college reading program was reported. At San Bernardino College the computer was being used in diagnosis, in prescribing reading instruction, and in end-of-unit evaluation in what appears to be the first CMI reading program ever developed at any level.[4]

In 1967, Hankin et al. reported their progress in developing an adult literacy program at Florida State University using vocationally-oriented reading materials.[5] Unfortunately the project was terminated before completion because of lack of funds. In that same year, Shanner explained PLAN (Program for Learning in Accordance with Needs), the first highly-publicized and widely marketed CMI program for reading instruction.[6]

In 1968, Green told the International Reading Association conference about his success in teaching words and letters to disadvantaged preschoolers. Boys were found to profit from the CAI more than girls.[7] Also in 1968, Coulson reported to the Southwest Regional Laboratory for Educational Research and Development on progress on the development of an instructional management system (eventually called AIMS).[8] By the end of 1968, computers had been tried or were being applied to reading instruction at nearly every level. Computers had been used for teaching preschoolers, primary-grade children, and adult illiterates. They were being used to manage the reading instruction of elementary and junior high students as well as the instruction in a college reading improvement course. However, the development of programs was terribly expensive, the hardware was not nearly as reliable as it is today, and computer installation was a difficult and expensive enterprise.

Nevertheless, in 1969 Butler reported on the first year of CAI reading in the New York City Schools. The drill-and-practice programs were purchased from Harcourt, Brace, and World, and hundreds of children used the 192 terminals each day in the sixteen schools in which they were installed.[9] The same program was in use at the same time in a number of Appalachian area schools served by the Eastern Kentucky Development Corporation,[10] while in Philadelphia, a four-school project (Project GROW) made 200 hours of CAI in reading available to 9th and 10th graders at eight terminals.[11] Also in 1969, Harris reported that the Boston Public Schools were gearing up a computer-based reading program designed to serve 11,000 children in learning reading and basic skills.[12] And while these huge school systems developed their large-scale projects, Way, working with one school in Monticello, Florida, developed a unique reading program she called DOVACK. In

DOVACK, children dictate stories which are typed into the computer and printed out for reading practice. Computer-stored word lists provide teachers with progress reports of words learned and tests of words learned recently.[13]

In 1970, an article in Reading Newsreport described the implementation of the Ransom CMI reading program in Fresno and Los Angeles,[14] while Waterford Township reported on its fourth and fifth grade CAI reading project.[15] Pfiel described the Migrant Student Record Transfer System (MSRTS) which was then operational in seven southern and western states. MSRTS was designed to enable school secretaries to call the nearest of eighteen area terminal operators and request the reading and math levels as well as health information about new migrant students.[16] And, balancing this large scale CMI project, was one at an elementary school for which students at a nearby high school programmed questions and reinforcing messages for on-line practice by elementary students after off-line reading of their basal readers.[17]

However, the big news in computerized reading in 1970 was project CARE (Computer Assisted Remedial Education) at the Pennsylvania State University. There the Cartwrights designed a program to train teachers of children who would become educationally retarded by the time they were nine or ten. After off-line reading of a specially-prepared text, teachers participated in CAI drill-and-practice activities at terminals housed in trailers trucked to the remote sites where they were employed.[18] A similar program was designed to teach phonics to teachers.[19] This project was the first major teacher-training project in reading instruction to be computerized.

By early 1971, most educators had learned that CAI was very expensive. The expense was not so much for computer time or memory, but for the creating of the lessons. A very early estimate was that one hour of CAI required 400 hours of preparation. A later estimate was that 254 hours were required to prepare one hour of CAI. However, more colleges were making computer facilities accessible to their faculties and students and money was available for innovative and experimental projects.

In 1971, Muller completed a master's thesis at Alberta by teaching preschoolers to read forty words which were pictured, labeled in print, and named aloud by machinery under computer control.[20] Also in 1971 Nashville State Technical Institute developed computer programs for scoring a comprehension-based informal reading inventory (a test on graded passages) and prescribing reading instruction for its students.[21] And in that same year, Lerner used computer simulation to train teachers in reading diagnosis.[22]

Through the preceding years (1964-1971) the Stanford project continued to develop its programs and to field-test them on an ever-widening clientele. In 1972, Fletcher and Suppes described the drill-and-practice reading program of the Computer Curriculum Corporation for children in grades four to six.[23] Atkinson and Fletcher described the program for grades one to three.[24] New CMI programs continued to be reported. In 1972, Randall described a CMI reading laboratory which was in operation at Christopher Newport College[25] and Beck and Bolvin described a CMI program which was to be developed to support the individually prescribed instruction of the Pittsburgh Learning Research and Development Center.[26] A somewhat different program resulted from the Kansas City project reported by Taylor. This program was designed to help students who encountered difficulty in printed programmed materials.[27]

In 1973, Lilman reported that 12,000 Chicago pupils were on-line daily at more than 850 terminals where they studied reading, language arts, and mathematics courses developed by the Computer Curriculum Corporation.[28] This report confirmed the continuation of a trend noted in 1970. Since computer curricula were so expensive to develop, the large school systems began to buy or rent existing programs.

In 1973, Fletcher and Suppes reported the application of computer technology to teaching reading and mathematics to deaf students.[29] This was one of the last CAI reading reports from the Stanford Institute for Mathematical Studies in the Social Sciences. However, in the same year Elliott and Videbeck reported the first reading program to be developed by the PLATO project at the University of Illinois. The program was one designed to develop reading comprehension among students preparing to take the G.E.D. examination for high school equivalency certificates.[30]

In the next year, 1974, three new CAI programs were reported to be operational. Obertino reported the work of the PLATO Elementary Reading Curriculum (PERC) group.[31] Luyben reported a word-teaching program in which EMR children matched computer-controlled picture displays by pointing a light-pen to words.[32] Golub reported a CAI adult literacy project.[33] After ten years of development, CAI reading programs continued to span the age ranges from school beginners to adults. However, the new projects all did zero in on beginning reading. Little effort seemed to have been made to develop advanced reading programs for teaching such things as critical comprehension or study skills.

More effort was being expended in developing programs for computer-managed reading instruction. Farber reported the development of the CMI program in reading for the Philadelphia Schools.[34] Belt and Skubal detailed methods for implementing a computer-managed system for the Wisconsin Design for Reading Skills Development, a reading program developed by the Wisconsin Research and Development Center for Cognitive Learning. They called their program WIS-SIM (Wisconsin System for Instructional Management).[35] Also in 1974, Dunn reported the tryout of a CMI reading program in Montgomery County, Maryland,[36] and the Capital City Career Center reported a computerized management system in which vocational students' basic skills needs tests were computer-scored allowing computer-written prescriptions to be derived.[37]

In 1975, Venezky and his colleagues developed the rudiments of a computerized on-line diagnostic reading testing program[38] and Eller described a computer-controlled educational television network operating in Fremont, California.[39] Handler revealed a CMI reading program in operation at Mineola (N. Y.) High School.[40] Roe and Aiken described the computer simulation of the construction and administration of an informal reading inventory.[41] And in that year Learning Unlimited started promotion of its CMI program called LMP (Learning Management Program) and bearing small resemblance to its ancestor, PLAN.[42] (As of 1977, both were still being marketed to public schools.)

Another pairing of computer control with educational television in a system for teaching reading in a structured system was described by Herlin, Bance, and Hansen in 1976. They had used TICCIT (Time-shared, Interactive, Computer-controlled Instructional Television) to teach a required freshman course in critical reading at Brigham Young University.[43] This was one of the first computerized reading programs to teach advanced reading skills with CAI and it remains one of the few to do so. CMI programs continued to proliferate at all levels. In 1976, the Economy Company bought out ACEPT, a diagnostic-prescriptive computerized testing program for young children,[44] and the Norfolk (Va.) Schools were reported to be using a reading and mathematics CMI program known as MIFS (Multi-ethnic Information Feedback System).[45] The pairing of instructional television with computer control was used to help homebound students in New York, where one hundred such students could use their phones to dial lessons for presentation on home video screens.[46]

In 1977, Obertino reported the progress being made by the PLATO Elementary Reading Curriculum project at the University of Illinois. The lessons shown were largely devoted to the initial stages of reading.[47] Seiler added a report on reading lessons developed by the University of Delaware's PLATO Project. Some of the lessons were for remedial reading and some were for children moving beyond the first stages of learning to read.[48] Another beginning reading program for handicapped youngsters was reported by Geoffrion and Bergeron of the University of New Hampshire,[49] and Goldberg and Rubin provided an overview of a modification of the TICCIT program (TICCIT + 10) useful in teaching reading and other subjects to deaf students.[50] The Dallas Schools were reported by Phi Delta Kappan to have developed a CAI beginning reading program for bilingual students.[51] Two CMI reading programs were reported by public school systems, Rapides Parish (La.)[52] and Fresno (Cal.).[53]

The number of reported new programs continues to increase each year. Both the publishers of reading materials and the administrations of school systems are developing CMI programs in reading. The strengths of such programs are in the constant monitoring of the skills tested as part of the management system. One problem is that other (less easily measured) skills are frequently overlooked because the management system does not provide tests for them. Another problem is that the ordering of the administration of tests may not be synchronized with the order of skill instruction in the school's reading program. When these two problems are recognized and compensated, the school system is usually very pleased with its CMI reading program.

CAI programs in reading are generally of two types, those commercially prepared programs purchased or leased by large school districts or those new programs designed for special pupil groups by college or university personnel or school-district personnel funded by grants. The commercially-prepared programs will probably proliferate because they are considered to be rather successful. The small innovative programs for special groups of pupils will come and go, depending upon the funding tendencies among granting agencies. Perhaps the greatest contribution that could be made to the future of computer-assisted reading instruction is the collection and storage of programs produced under grants and the conversion of these programs for delivery by the newer computers, so that future developers could make use of the tremendous efforts of those who have preceded them.

1. Atkinson, R. C., & Hansen, D. Computer-assisted instruction in initial reading: The Stanford Project. Reading Research Quarterly, 1966, 2, 5-25.

2. Wilson, H. A., & Atkinson, R. C. Computer-based instruction in initial reading: A progress report on the Stanford Project. Stanford, California: Institute for Mathematical Studies in the Social Sciences, 1967.

3. Suppes, P., Jamison, D., & Butler, C. Estimated costs of computer-assisted instruction for compensatory education in urban areas. Educational Technology, 1970, 10 (4), 49-57.

4. Williams, G. The use of the computer for testing, programming, and instruction. Paper presented at the National Reading Conference, St. Petersburg, Florida, December 1966.

5. Hankin, E. K., Smith E. H., Smith, T., & Ritchie, M. The development of pre-vocational education literacy courses for use with computer-assisted instruction of disadvantaged youth and adults. Technical Progress Report Number BR-6-1458-PR1, Tallahassee, Florida: Computer-assisted Instruction Center, Florida State University, 1967.

6. Shanner, W. M. PLAN. A system of individualized instruction utilizing currently available instructional materials. New York: Westinghouse Learning Corporation, 1967.

7. Green, D. R. Learning to recognize words and letters on a CAI terminal. Paper presented at the Annual Conference of the International Reading Association, Boston, April 1968.

8. Coulson, J. E. Progress report for the instructional management system: A report to the Southwest Regional Laboratory for Educational Research and Development. Santa Monica: Systems Development Corporation, 1968.

9. Butler, C. F. CAI in New York City - Report on the first year's operation. Educational Technology, 1969, 9 (10), 84-87.

10. Mynhier, B. The impact of federal programs on learning to read in Appalachia. Paper presented at the Annual Conference of the International Reading Association, Kansas City, April-May 1969.

11. Charp, S. A., & Wye, R. A. Computer-assisted instruction in a large school system. Journal of Educational Data Processing, 1969, 6 (1), 28-39.

12. Innovative implementation of computer-aided instruction. In L. A. Harris (Bibliographer) Title III Pace Projects in Reading. Vol. 2. Bloomington, Indiana: ERIC/CRIER Reading Review Series, 1969.

13. Way, F. L. The DOVACK model. Educom, 1969, 4 (5), 6-7.

14. Serlin, J. Two for one. Reading Newsreport, 1970, 5 (1), 18-22.

15. Hannan, T. Computer assisted instruction--State of the art summary. Audio Visual Instruction, 1970, 15 (6), 93-94.

16. Pfiel, M. P. New data processing network tracks child's full scholastic and health history. American Education, 1970, 6 (5), 4-6.

17. Serlin, J. Good grief! Charlie Brown's computerized! Reading Newsreport, 1970, 4 (6), 29-30.

18. Cartwright, G. P., & Cartwright, C. A. Handbook for CARE 1--Introduction to the education of exceptional children. Laboratory Report Number R36. State College, Pennsylvania: Computer Assisted Instruction Laboratory, Pennsylvania State University, 1970.

19. Thompson, R. L. Computer-assisted phonic analysis: A validation study. Laboratory Report Number R52. State College, Pennsylvania: Computer Assisted Instruction Laboratory, Pennsylvania State University, 1973.

20. Muller, V. A case for early reading. Unpublished M.Ed. thesis. University of Alberta, 1971.

21. Wares, M. B. Use of the computer in individualized remediation at Nashville State Technical Institute. Paper presented at the National Reading Conference, Tampa, Florida, December 1971.

22. Lerner, J. W. *Computer simulation: A method for training educational diagnosticians.* Paper presented at the annual conference of the International Reading Association, Atlantic City, April 1971.

23. Fletcher, J. D., & Suppes, P. Computer-assisted instruction in reading: Grades 4-6. *Educational Technology*, 1972, *12* (8), 45-49.

24. Atkinson, R. C., & Fletcher, J. D. Teaching children to read with a computer. *Reading Teacher*, 1972, *25* (4), 319-327.

25. Randall, A. F. A computer-managed reading laboratory program. *Reading World*, 1972, *11*, 286-295.

26. Beck, Isabel L., & Bolvin, John C. A model for non-gradedness, the reading program for individually prescribed instruction. *Elementary English*, 1969, *46*, 130-135.

27. Taylor, S. S. *A comparison of two public school computer assisted instruction projects.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, April 1972.

28. Lilman, G. H. *CAI in Chicago.* Paper presented at the annual meeting of the Association for Educational Data Systems, New Orleans, April 1973.

29. Fletcher, J. D., & Suppes, P. *Computer-assisted instruction in mathematics and language arts for the deaf.* Final report. Stanford, California: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1973.

30. Elliott, P., & Videbeck, R. Reading comprehension materials for high school equivalency students on the PLATO IV computer-based education system. *Educational Technology*, 1973, *13* (9), 20-22.

31. Obertino, P. The PLATO reading project: An overview. *Educational Technology*, 1974, *14* (2), 8-13.

32. Luyben, P. D. *The effects of pictures on acquisition of a sight vocabulary in rural EMR children.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans, February 1973.

33. Golub, L. S. A computer assisted literacy development program. *Journal of Reading*, 1974, *17* (4), 279-84.

34. Farber, I. J. *The development and implementation of a multi-level management information feedback system.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, February 1974.

35. Belt, S. L., & Skubal, J. M. *Individually guided education and the Wisconsin system for instructional management (WIS-SIM).* Paper presented at the conference of the Association for Educational Data Systems. New York, May 1974.

36. Dunn, A. (Ed.). *Computer assisted instruction program. A three year report covering July 1, 1971 through June 30, 1974.* Rockville, Maryland: Board of Education, Montgomery County Schools, 1974.

37. Danford, J. CMI at Capitol Area Career Center, Mason, Michigan. In H. E. Mitzel (Ed.) *An examination of the short range potential of computer managed instruction: Proceedings of a conference.* Chicago, November 1974. (ED 112-943)

38. Venezky, R., Bernard, L., Chicone, S., & Leslie, R. *On-line diagnosis of reading difficulties.* Technical report number 327. Wisconsin Research and Development Center for Cognitive Learning. Madison, Wisconsin: University of Wisconsin, 1975.

39. Eller, L. M. Individualized learning using T. V. *Educational Broadcasting*, 1975, *8* (4), 27-30, 42.

40. Handler, H. The computer comes to the English classroom: Computerized monitoring of pupil progress in reading. *Educational Technology*, 1975, *15* (11), 34-35.

41. Roe, M. H., & Aiken, R. M. A CAI simulation program for teaching IRI techniques. *Journal of Computer-based Instruction*, 1976, *2*, 52-56.

42. *Learning management program: Management training manual.* New Cannan, Conn.: Learning Unlimited Corp., 1975.

43. Herlin, W. R., Bance, G., & Hansen, D. M. *Teaching college level reading in a computer assisted-computer managed setting*. Paper presented at the annual conference of the International Reading Association, Anaheim, May 1976.

44. Yawkey, T. D., & Lewis, L. *ACEPT: Assessing children for early prescriptive teaching*. Oklahoma City: Center for Educational Assessment, The Economy Company, 1976.

45. Tokar, E., & Ford, V. *Effectiveness of an ESAA funded information feedback system on the reading and arithmetic achievement of black and white fifth grade pupils*. Paper presented at the annual meeting of the American Educational Research Association, San Francisco, April 1976.

46. Computer-tutor. *American Education*, 1976, *12* (5), 4.

47. Obertino, P., Fillman, L., Gilfillan, J., Silver, D., & Yeager, R. *Elementary Reading on PLATO IV*. Urbana, Illinois: Computer-based Education Research Laboratory, University of Illinois, 1977.

48. Seiler, B., & Weissman, J. *Remedial reading on a computer called PLATO*. Paper presented at the annual convention of the Diamond State Reading Association, Dover, Delaware, November 1977.

49. Geoffrion, L. D., & Bergeron, R. D. *Initial reading through computer animation*. Paper presented at the annual meeting of the American Educational Research Association, New York, April 1977.

50. Goldberg, L. M., & Rubin, F. S. *An overview of the TICCIT + 10 computer system at the Model Secondary School for the Deaf*. Paper presented at the annual meeting of the Convention of American Instructors of the Deaf, Los Angeles, June 1977.

51. Students use talking computer in Dallas bilingual program. *Phi Delta Kappan*, 1977, *59* (4), 234.

52. *Individualized reading instruction for students computer-managed reading system: Validation report*. Alexandria, Louisiana: Rapides Parish School Board, 1977.

53. Lister, W., & Williams, J. *Fresno reading program with computer assistance*. Paper presented at the Rocky Mountain Regional Conference of the International Reading Association, Salt Lake City, November 1977.

A UNIVERSAL DISSEMINATION FOR COMPUTER PRESENTED COURSEWARE

Rick Meyer
Academic Computing Association

Writing courseware for Computer Presented Drill & Instruction is an extremely time consuming process. Many computer manufacturers have a clearinghouse which distributes courseware in a machine readable form. This is a service they provide to their own users. But due to differences in software and hardware, the machine-readable courseware from one manufacturer's computer usually cannot be read or run on another manufacturer's computer.

The need to disseminate CPD&I courseware is obvious. The problems associated with disseminating it in a form which is machine readable by most computers are overwhelming. As a consequence, the Academic Computing Association has been founded as a non-profit corporation to serve the educational community as a clearinghouse for CPD&I courseware. Initially, contributed courseware will be distributed in human readable form.

This presentation at the 16th Annual AEDS Convention follows a brochure copyrighted by the Academic Computing Association entitled "Computer Presented Drill & Instruction." It answers the following eight questions.

1. What is CPD&I?
2. Who writes CPD&I courseware?
3. What is an Author Language?
4. Do many computer manufacturers support an Author Language?
5. Is there one Author Language which runs on most computers?
6. Currently, how can CPD&I courseware be disseminated?
7. Who can contribute CPD&I courseware for dissemination?
8. How will information on available courseware be publicized?

Copies of the brochure will be made available at the presentation. In addition, the Academic Computing Association publishes a newsletter entitled Courseware Quarterly. Write to P. O. Box 27561, Phoenix, AZ 85061, for additional information.

BEYOND INDEPENDENCE: A VIEW OF GAMES AND PERSONAL COMPUTING

Kenneth L. Modesitt, Visiting Associate Professor, Computer Science Department
California State University Northridge, Northridge, CA 91330 (213) 885-3398

Personal computing and interactive computer games are presented as two concepts which have a great deal to offer one another. The rationale for each is discussed, stressing the importance of the feeling of independence for the personal computer user. An extensive list of games, competitive and cooperative, is given as an initial attraction to draw many individuals into the home computer market. Many of the games and other interactive applications were created by people who had grown weary of playing Startrek for months on end. It is suggested that other personal computer users will also tire of playing games solely, and will begin to create their own processes. Based on the present, it appears the future will see considerable sharing of efforts among users of personal computers, perhaps within the framework of a network.

Introduction

Independence. What overwhelmingly deep and warm (sometimes scary) feelings are evoked from most people when this word is spoken in sincerity. It is my opinion that this value, in conjunction with the technological miracle of the microcomputer, is largely responsible for the revolution we are seeing today in computing. The other trait which contributes to the revolution of personal computing is that of curiosity, and is most obvious in the initial attraction of a person to an interactive computer game. In this paper, I will discuss the interaction between games and personal computing, why we have them now, including an extensive list of the former, where we may be going from here and why I hope to see a growth beyond the strongly-held value of independence to that of an interdependent community of personal computing users.

Why Do We Have Them?

Games. Games are fun. That's hardly a surprise-we've known they are fun since we were old enough to play our first one. Perhaps games, play, and fun are all synonymous. They pique our curiosity, and encourage us to explore uncertain situations without fear. Computer games permit one or more persons to play each other over distances using the machines. They will play with us alone in case sister Sue or uncle Bob is unwilling or unable to play right now, and the diversity and complexity of the games is remarkable. There are other, perhaps more pedantic reasons, why we are attracted to games on computers. They allow us to develop our problem--solving skills in a non-threatening environment. Many of us are anxious to demonstrate our superiority over the computer, the clock, or other players. This competitive aspect is very strong in many (most?) games.

Personal Computers. A person in the United States today who does not know something about computers is either illiterate or lives in isolation. The impact of computers is felt nationwide and frequently worldwide. Yet many (most) people feel alienated and threatened by such machines. This alienation may be on the verge of breaking down, in my opinion thanks to the ingenuity of microcomputer designers and the marketing skill of businesses, small and large.

The advent of the personal computer in retail outlets nationwide for a price comparable to a good color television set or stereo console is a major step in the revolution of bringing computer power to the people. When Joan or Joe Smith have their own personal computer, they are unlikely to be as intimidated as now when they are told "It's the computer's fault" by Sears, Mastercharge, Consolidated Edison, California State University, United Airlines, Union Bank, General Motors, the property tax assessor, Department of Motor Vehicles or the Social Security Administration.

A personal computer signifies that an individual is no longer solely dependent upon what he or she is told by the computer belonging to a large, seemingly insensitive institution. And, to no one's surprise, people like that. To be more independent of big business, big government, and big education is a value sought tirelessly by many people. The computational power once available only to big establishments for millions of dollars is now readily available to most of the general population for roughly $1,000, and the price continues to decrease.

The Hook

We are all familiar with the story of how the staff at the carnival side-shows would secure their customers. To feel oneself being dragged to the entrance by a hook caught in a collar or sleeve is a powerful inducement to attend the show.

The moral is clear - just the availability of carnival marvels or technological marvels such as the personal computers discussed previously does nothing to guarantee customer response. You need a hook. Well, the burgeoning individually owned or franchised computer stores, and soon the large retail outlets, have just a "hook". It is called "playing games on the computer". This technique is a superb way to attract a customer initially, given a near universal propensity for games. And, of course, the larger and more diverse and more complex the stock of computer games, the more interest will be generated by the customer, soon to be a purchaser, of her or his own personal computer. There is also a substantial portion of buyers who have additional reasons; we shall return to this class in the next section.

In this section, I will list some of the games which I have used as a "hook" in introductory computer science classes for several years. In my opinion, they would be ideal to implement on personal computing systems (several have been) as they are interactive, written in Basic (95%), and extensively tested. It will be obvious that most of the ones listed under games have been published in Creative Computing, Personal Computing, Interface Age, Byte, Kilobaud, People's Computers, etc. A few come from other sources (6), some from myself and many were written by students who had been "hooked". As most computer science faculty are aware, there are always a few students who are exceptionally clever, witty, and thorough in creating interactive programs, several of their term projects are included in the list. The entries are taken verbatim from a larger set of program descriptions handed out to students at the beginning of the term.

Program Names and Descriptions

(All programs are interactive, unless otherwise noted.)

Applications

Name	Description
220GRA	Determines grade as function of units passed, final and project
220NEWS	Message file for CS220
BUDGET	Derives a family budget in an iterative fashion
CRONY	Introduction to KRONOS system
CSGRADE	Instructor Grading for CS220
GENES	Genetic Simulation (not interactive)(Fortran source is in unit record room by computer center of IPFW)
LINEAR	Linear Regression
LYNN	Projectile Physics "Game"
MATHREV	Review of elementary arithmatic for junior high level.
SCABASIC	Structured program preprocessor for basic
SIMULT	Simultaneous Equations
TREES	Tree identification
Personal Data Bases and Questionnaires	
DTBS	CS220 Project on personal information
FUTURES	Delphi questionnaire on computer impact
FUTURAN	Analysis of responses from FUTURES
QUERY	"Interviews" beginning students in personal fashion, letting them ask questions.
SEC	Discussion of computer security and privacy issues
SCABBYS	CS220 project for keeping records on players of star trek game
SOCIETY	Questionnaire from Creative Computing on computer impact
ULTIMAT	CS220 Project for privacy of personal data bases
Simulations and Models	
2001	A student's view of his day in the year 2001
WORLD	Extensive technologically optimistic simulation of world (Forrestor's model)
WORLDM	Simple world simulation model (graphic output)
Games	
BAGELS	Number Guessing
BATNUM	Varient of NIM-Battle numbers
BALLRUN	Simulation of Civil War battles
DAMON	Computer Dating
DATE 4U	Computer Dating Matches (not interactive)

DAYTONA	Choose car type and racing sytle
DODGEM	Try to cross a board before opponent
DRAG	Design your own dragster
FTBLL	Extensive football game
GAL	Command Galileo shuttlecraft of Starship enterprise
GAMES	Extensive index to games
HAMURBI	Govern ancient Sumaria
HANGMAN	Word Guessing
ICBM	Intercontinental missile game
INDEX	Major index for games, programming lessons, notefiles, etc.
LOST	Cooperative game of being lost on gold-filled island
MADLIB	Lets group of users make up riotous stories
MAGIC	Game to draw magic square
MONSTER	Don't get eaten.
OLDMAID	Classics card game
RACE	Choose your car in road rally
REVERSE	Sorting game with numbers
RIDDLES	Just what it says.
SLOTMAC	Win a fortune on a slot machine
SPILL	Spill and Spell game to make words from random letters
STRBS	Protect your starbase
TREK	Star Trek
WUMPUS	Hunt the wumpus in a weird scary cave
XB70	Star Trek

Graphics

GARYPIC	Portrait of Gary
LOVE	Robert Indiana's love poster
PIC	Draws arbitrary picture
SINE	Sketches sine curve
SNOPIC	Snow and mountain picture
SURFACE	Three-dimensional plot of math function
WORMY	Watch a worm chew up a garden
XMAS	Portrait of Mary and baby Jesus

For some time, I have been concerned with the emphasis placed on competition in several facets of education and computers (8,9). Even prior to compiling the above lists, a major point was obvious to me: playing a game and competing are not synonymous concepts. Yet how often do we treat them as the same. When we are asked to play a game, either by a person or a computer don't we automatically assume they mean a zero-sum game, i.e., if one person wins, the other loses? Yet if we go to the American Heritage Dictionary, we find the following:

> "game: 1. A way of amusing oneself a pastime; diversion 2. (Mathematics) set of rules completely specifying a competition... 3.A sport or other competitive activity governed by specific rules." (11)

Although games may be competitive, the concepts are not identical. In our world of today, I believe that the concept of cooperation, rather that competition, should receive increasing emphasis. The reasons for this desire to work with a community of responsible individuals toward a mutual goal are detailed elsewhere (8). Also given in the same paper is how cooperation has been included explicitly in several of my computer science courses. Summarizing, it was done in 5 ways;

1) study partners for the units of a personalized system of instruction (PSI) course.
2) mutual design and use of interactive programs assigned in the units,
3) use of cooperative exercises, computer and otherwise,
4) parties, and
5) design of a computerized personal database by a small group.

The results of point 5 above are seen partially in the extensive programs listed earlier under "Personal Data Bases" (DTBS, FUTURES, FUTURAN, SCABBYS, ULTIMAT) and "Games" (GAMES, INDEX). These are systems which one may use "to amuse oneself, as a pastime, a diversion", and yet are not solely competitive. They were designed and implemented cooperatively using reliable techniques such as structured design and programming, top-down design for games (5), and "egoless" programming (2). They are not the same as cooperative games, e.g., LOST or MADLIB. The latter class is very difficult for most of us to even conceive - can you give me the names of just five games which are based on cooperation? I'd be delighted if you could provide me a source for one. Compare the difficulty of that request with one for a source of our most popular (competitive) game-STARTREK. And yet, what role did competition really play in the original television show? As Douglas Crockford pointed out recently, the mission of the Enterprise was to: "Seek out new worlds and civilizations, and to go where no man had gone before", not to blast Klingons out of the galaxy (3).

Games for computers then are a useful "hook" to interest people and to provide an initial contact which, hopefully, is non-threatening to most. Competitive games can be fun, but we can also begin to offer games in which someone does not have to "lose" in order for someone to "win". Moreover, we have further evidence that people can design and implement fun type programs in a cooperative manner. This will become increasingly important as we move further along in the personal computing revolution

Where Do We Go From Here?

For sometime now, I have been struck with the remarkable analogy of how an individual relates to his or her private automobile, and what the future may portend for how an individual relates to his or her personal computer. When we acquire our first car, how much of our time with it is spent in games, particularly competitive ones, e.g., purchase price, 0-50 time, miles per gallon (rare), engine size, interior, paint job, mags? Sure, we use it in a cooperative fashion once in a while, e.g., taking the other kids to school, but was that the real reason we purchased it? And then, after the purchase, a few of us are actually sufficiently motivated and clever enough to open the hood and build or rebuild parts here and there. Some people continue this interest for most of their life, through one car after another. Yet most of us are content to let someone else maintain and repair the vehicle after a short time. The private car as a technological miracle of the early 20th century is a superb example of independence, or at least semi-independence, as the recent oil boycotts have shown. The pervasiveness and widespread distribution throughout our culture of this product constitute a ringing testimony to our belief of going where we want and when we want. It is truly a remarkable extension of our ability to walk. Such a concept of independence is certainly not lost on the more centralized carriers of people, e.g., railroads, planes, busses: just note their "freedom fares" or the "go anyplace, anytime fares."

It would not be difficult for anyone familiar with personal computing to rewrite the above paragraph, replacing automotive references with personal computing ones. The analogy is striking. We become hooked on computer games (most of us). We purchase our machine and play Startrek for six consecutive months, perhaps taking time out to use it for pre-programmed mundane tasks such as family budgeting. A few of us (many of us right now because of the relatively small population) actually tinker with the insides adding a CPU (engine) here, putting on paddle controllers or voice synthesizers (4 barrel carbs or mags) there. I certainly think the "tinkering" with microcomputers will contribute much more to future computer design that does current "tinkering" with cars for future car design. This is not an attempt to put down kit-builders or designers; it is an attempt to say I do not believe many people will do it, no more than most of us would do serious work on our car.

However, the analogy begins to break down at this point. For there is a vital distinction between a personal car and a personal computer. If you want to change the latter, you do not have to tinker with hardware. You can build yourself a new (virtual) machine by composing programs. Initially, I would not expect a large proportion of the owners of personal computers to take such a task upon themselves, unless of course, they knew about programming before. But as the hardware and particularly the system software improves, I expect the complexity of the user interface to decline. Although natural language is still not in the immediate future, there are processors available which are considerably easier than those you and I learned. Apple Corporation's version of Basic permits three-dimensional color graphics (15). Logic Machine Corporation's ADAM bears very little resemblance to the strict procedural languages known to most of us (7). By the time this paper is presented, other non-rigid user interfaces will certainly have appeared. And I would wager that many of these interfaces will have their origin with the owners of personal computers or institutions which are firmly committed to the concept of "Computer Power to the People."

The importance of making it easier for the personal computer owner to program will become abundantly clear to her or him after the six months of Startrek and running the few non-game programs from the cassette tape purchased along with the machine. Now, if the machine belonged to someone else, it might well cease to be used at this point. However, a person who has made a substantial investment will not be so likely to simply leave it turned off.

I believe personal computer owners will begin to generate useful programs, initially for their own purpose, but later for the neighboring community. Substantiation for this is not difficult. How many of us know students who eventually tire of playing games solely over the interactive terminals, and begin to write extremely clever and stimulating computer-based education (CBE) lessons? Witness the popularity of the "Lemonade Computer Service Company" articles in Personal Computing (12), whose intent is to show how home computers can be used for small businesses ("lemonade stands and paper routes"). In the first issue of ROM (14), Hesh Wiener relates how he was able to turn his teletype into braille-printing terminal in six hours, and how his trouble getting his blind friends off his computer. Although hardly a personal computer in price, PLATO provides an excellent example of how former game-playing students now write some of the best CBE lessons (10). The programs listed in section III included many written by students, both of a game nature (FTBLL, OLDMAID, BATNUM, SLOTMAC, SPILL), and those which are application-oriented (CRONY, CSGRADE LINEAR, LYNN, MATHREV, SIMULT, TREES.) All of these were written by students on interactive systems owned by a university, i.e., they had to compete with hundreds of other users even to use the terminals. It is exciting indeed to contemplate what may happen when many of the same students, as well as many of us, have virtually unlimited access to personal systems. Where we are going is, of course, anyone's guess. I believe that providing increasingly better human interfaces, languages and hardware, on personal computers will reap enormous benefits in terms of application programs for use by the individual user and the community at large. This "equalizer" a term from the pages of Personal Computing, finally begins to offer individuals some of the remarkable symbol-processing power available to large institutions for the last twenty years. The amplification of some of our cognitive abilities offered by personal computing will, I believe, be at least as significant as the amplification of some of our locomotion abilities offered by personal cars.

And Finally, How Do We Go From Here To There?

The more availability of application programs on personal computers will do very little to guarantee that they will be used. The identical point was made earlier with regard to simply making microcomputers available. Consequently, personal computer owners who have written programs for the use of their family only will most likely be prodded, either internally or by their neighbors, to make their computer facilities more widely used. This problem of distributing a product is an old one, but usually so for businesses; individuals do not have much experience with the concept as a supplier of the product, only as a recipient.

There is a distinctive quality about the products which are now distributed by people involved in personal computing. An enormous amount of it is free, or virtually free. The quality of interdependence: "I created something; I'll share it with you; perhaps you'll share something with me" is one which underlies many interchanges of knowledge and/or products. We realize fully that asking a high price for our product will prohibit its distribution. So the concept of mutual cooperation is one that is everywhere evident in personal computing. I believe it is an excellent and appropriate model in comparison to the competitive one with which we are too familiar. And it is a natural growth from the model of strict independence. This same transition from competition to independence to interdependence is one traced in education in an earlier reference (8).

Now sharing one's efforts is oftentimes easier said than done. There are currently at least four major routes used:

1. Publications. Send a listing of your program and associated documentation to a magazine publisher. Many of the magazines are listed in the bibliography.
2. Removable storage media. Send or sell cassette tapes or floppy disks to a person requesting a program. Most computer stores do this. See also the September, 1977 issue for Interface Age (13).
3. Centralized time-sharing systems. Give the requestor the password to use your program stored in a centralized data base. PLATO in an excellent example of this.
4. Networks. Send your program over communication links to another computer or let the remote user access your program. The ARPANET is an obvious example.

Each method has its own advantages and disadvantages, but one of the most serious concerns in all is portability (is probably of least concern in type 3). Give the predisposition of personal computer owners for independence and the ready availability of the telephone network, I strongly imagine alternative 4, networks, will become increasingly common. The Personal Computing Network (PCNET) started in the Palo Alto, California area is an excellent example (1). Of course, these communications lines are for computers. Also of vital importance is the communication lines between personal computer owners and people in the community having no such machine. The latter is particularly exciting in my opinion. Peoples Computer Company in Menlo Park, California staffs a Community Computer Center to bridge just such a gap (4).

Summary

In conclusion, I have tried to indicate why I believe the advent of the personal computer portends a true revolution for much of our culture. There is an enormous appeal for a technological product when it can be utilized extensively by an individual person or family unit. Witness the popularity of calculators, cars, televisions, radios, houses, and solar energy collectors, to name a few. Most people are extremely interested in artifacts which allow them to decrease their reliance on large institutions, i.e., those items which seemingly increase the abilities and independence of the owner. I have indicated that computer games of all types; competitive. cooperative, and application could well be used as they have been, to increase the interest in personal computing. However, it appears that games are not sufficient stimulus over the long-term, and so I believe personal computing users will grow beyond games. They will become increasingly interested in and capable of constructing their own processes, some games and some applications. The literature indicates this is already occuring. Some of the resulting programs will be used primarily for the needs of the owner, but I believe a considerable amount of the effort will be shared with other personal computer owners. This sharing has occured initially by publications and removable storage media, but networks will probably appear with increasing frequency. Moreover, there will be considerable sharing between those who own the machines and those who do not, yet. I fully expect the former class to grow dramatically in the next decade. It promises to be an exciting one for us as an interdependent community of individuals.

References

1. Caulkins, Dave. "PCNET : 1979", "People's Computers, Volume 6, Number 2, Sept.-Oct. 1977, pp. 18-20

2. Cheney, Paul. "Egoless Programming: As an Instructional Technique." AEDS 14th Annual Convention, 1976, pp. 222-224

3. Crockford, Douglas. Letter to Editor Personal Computing, Volume 1, Number 4, July/August, 1977. p.8.

4. Daily, Dean. "A Day in the Life of Community Computer Center." People's Computer Company, Volume 5, Number 4, Jan.- Feb., 1977, pp. 8-9.

5. Guthery, Scott B. "Top-Down Design of Computer Games." Personal Computing, Volume 1, Number 3, May/June, 1977, pp. 124-128.

6. Hicks, Bruce. "Computer Outreach" Computers and Society, ACM Special Interest Group, Volume 7, Number 3, Fall, 1976, pp. 10-14.

7. "Logical Machine Corporation ADAM." Datapro Reports on Minicomputers, September, 1976. Report M11-587,101.

8. Modesitt, Ken. "A Community of Individuals Cooperation and Individualization in Computer Science Education." National Computer Conference, 1977, pp. 561-567.

9. Modesitt, Ken. "The Tangled Triangle: Cooperation, Computer-Based Education and Personalized Systems of Instruction." ADCIS Summer Conference, 1976. pp. 320-324.

10. Modesitt, Ken. "Faculty-Student Development of Computer-Based Instructional Modules on PLATO IV." Office of Instructional Services, Measurement and Research Center, Purdue University, August 1975.

11. Morris, William (editor). The American Heritage Dictionary of the English Language, 1969.

12. Personal Computing, Nels WinklessIII (editor), Benwill Publishing Corporation, Brookline, Mass.

13. Shamburger, Bud. "General Ledger Program." Interface Age, Volume 2, Issue 10, September, 1977, pp. 26-56

14. Wiener, Hesh. "Charity-Begins-at-Home Computing." ROM, Volume 1, Number 1, July, 1977, pp. 19-25

15. Wozniak, Stephen. "The Apple-II BYTE, 1977

A REALISTIC USE OF COMPUTERIZED METHODOLOGY IN ACCOUNTING INSTRUCTION

Roy James Parrish, Jr.
Henry Jefferson Beck
Danville Community College
Danville, Virginia

The widespread use of credit by individuals and businesses has increased the number of business transactions which must be recorded, classified, and summarized. In addition various governmental regulations have required business organizations to provide accurate accounting information in great volume and at tremendous speed. Students who plan to be successful in the business world must be prepared to deal with these demands. The demands placed on accountants to provide this information within a limited time require that some method, in addition to a manual system, be used to process accounting transactions. The computer has proven to be a satisfactory addition to use when teaching accounting for use in the modern business world. A series of computer programs developed at Danville Community College introduces the student to the problem-solving capability of the computer while reinforcing his knowledge of the accounting cycle. The programs are designed to eliminate the tedious, time-consuming posting and balancing function encountered in the accounting cycle. The programs also provide the student with a complete set of accounting records that show the relationship of the various parts of the accounting cycle.

Introduction

To say that society is engaged in a continuing technological revolution is indeed an understatement. Verification of this fact is provided by the myriad new devices and concepts with which we are confronted day after day. The computer is serving as the means for many of these technological enhancements which are currently taking place.

There are few people who would disagree with the observation that the computer has significantly influenced almost all aspects of society including education. In colleges and universities throughout the world, the use of the computer has become an important aspect of the instruction presented to students in many courses. While this statement is true for many subject areas, it is not true for the Principles of Accounting course. The method of instruction in the Principles of Accounting course in most colleges has changed very little during the past 20 years. Instruction consists of lecture by the instructor and the manual solution of problems by the student. Accounting instruction must begin to reflect the state of technology which exists in the business world. Students enrolled in the Principles of Accounting course must be introduced to the use of the computer as a tool of the accountant at the time they are receiving instruction in basic accounting concepts if the student is to gain a realistic view of the work of the accountant.

Previous Attempts at Incorporation of Automated Accounting Concepts into Principles of Accounting

Accounting instructors have made attempts to incorporate automated concepts into the Principles of Accounting course. These attempts can be categorized into basically three methods or approaches of instructing accounting students concerning computers. While these approaches to integration may be preferable to total exclusion of computer concepts, none presents the student with an accurate picture of the computer as a working tool of the accountant. These approaches are:

The student reads about how computers are used in accounting. Using this method, the student is instructed to read a chapter or chapters in his accounting textbook which describes the role of the computer in the accounting cycle. It is very likely these chapters will contain highly complex systems flowcharts which when examined leave the student with nothing more than a headache.

The student writes programs to perform

accounting tasks. Some instructors have attempted to integrate computer concepts into the Principles of Accounting course by requiring students to write computer programs to solve accounting problems. In some courses, students have been expected to design and program sophisticated accounting systems in a single term. This approach is totally unworkable as the beginning accounting student usually lacks both the programming and accounting expertise necessary for success.
The student plays a game which is run on a computing system. Several packages of programs have been marketed which do not require the student to demonstrate a mastery of accounting concepts for use. Students simply key a group of numbers which when inputed produce a listing. These packages in no way depict the integral role of the computer in business.

Reasons for Failure to Incorporate Automated Applications

The major factor responsible for the failure to integrate the realistic use of the computer into accounting instruction can be attributed to the resistance to change of accounting faculty. This resistance to change, while a natural human emotion, must be overcome if today's accounting students are to receive instruction that will prepare them to function in a profession which relies extensively on the computer. Factors listed below often contribute to resistance:
Lack of experience or lack of recent experience. Many accounting instructors fail to see a need for inclusion of computerized methodology in the Principles of Accounting course. This view can frequently be attributed to the background of the instructor. The accounting instructor with limited or no practical experience in the business world may see little need for teaching automated procedures since he has never experienced a need for this knowledge. The same may be said for the accounting instructor who has not worked as an accountant in recent years.
Lack of training. The accounting instructor may see the need for a knowledge of automated procedures by students but hesitate to include instruction in his classes because he does not have a thorough knowledge of these procedures himself. This may be a particularly acute problem for the instructor who does not have practical work experience.
Not enough time. Some instructors will state that there is not sufficient time to cover all material currently being presented in the Principles of Accounting course and also present additional instruction regarding the use of the computer. They are absolutely correct if all material is taught using a manual system and *retaught* using a computerized method. An examination of the most widely used accounting text will reveal the presentation of materials in a manner which does not reflect current accounting methods. The Principles of Accounting course should be revised--eliminating useless or outdated material and replacing this material with appropriate computer applications.
Lack of equipment. Proper equipment may not be available for use in accounting instruction. It is up to the faculty to ensure that the administration understands the need for equipment. Could a viable secretarial science program operate without typewriters?
Lack of suitable instructional materials. Until recently, instructional materials provided by publishers did not provide a realistic exposure to the use of the computer in processing accounting transactions.

Development of Instructional Materials Used to Incorporate Automated Accounting Concepts into Principles of Accounting

The accounting department and data processing department of Danville Community College undertook the development of an instructional accounting package that could be used to integrate computer concepts and automated procedures into the Principles of Accounting course. Development began with these assumptions:

1. The Principles of Accounting student does not need to be able to write computer programs.
2. Materials developed should demonstrate all phases of the accounting cycle.
3. Materials developed should reflect actual accounting practice and be compatible with most Principles of Accounting textbooks.
4. The student should be required to demonstrate a mastery of accounting concepts.
5. Data entry and error detection should be made as easy as possible.
6. Accounting instructors using the package should not need to know how to write computer programs.

With these assumptions as a basis for development, a package of learning materials consisting of coding forms to be used in the recording of accounting transactions, a package of FORTRAN IV computer programs capable of producing a series of accounting documents and a workbook consisting of a group of accounting problems was developed. The package of programs consisted of two groups of programs referred to as PHASE I and PHASE II. PHASE I programs accomplished the recording, summarizing, and posting

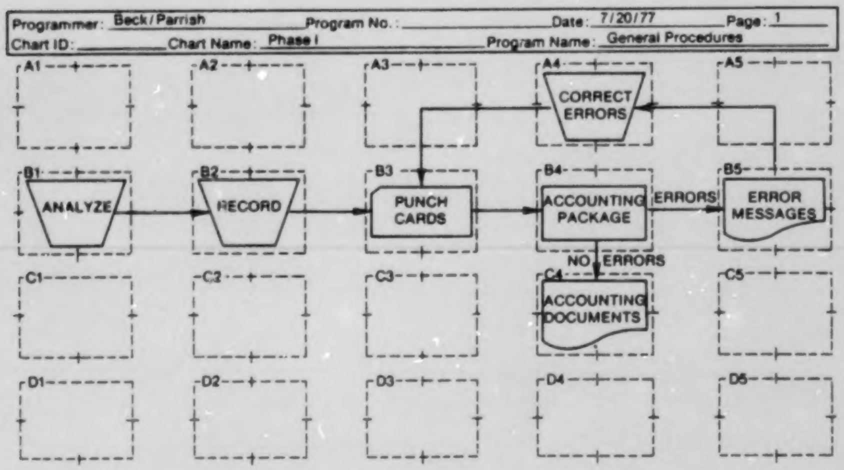

Programmer: Beck/Parrish Program No.: Date: 7/20/77 Page: 1
Chart ID: Chart Name: Phase I Program Name: General Procedures

Figure I

functions of the accounting cycle while PHASE II programs performed the adjustment and financial statement preparation functions of the accounting cycle. A sales journal, purchases journal, cash receipts journal, cash payments journal, general journal, general ledger, and trial balance are produced during execution of PHASE I. Adjusting entries are recorded in the general journal, posted to the general ledger and an adjusted trial balance, income statement, capital statement and balance sheet are generated by the PHASE II programs. The package provided the capability to solve accounting problems for sole proprietorships, partnerships, and corporations as well as the capability to establish beginning account balances.

Incorporating Computerized Concepts into Classroom Instruction

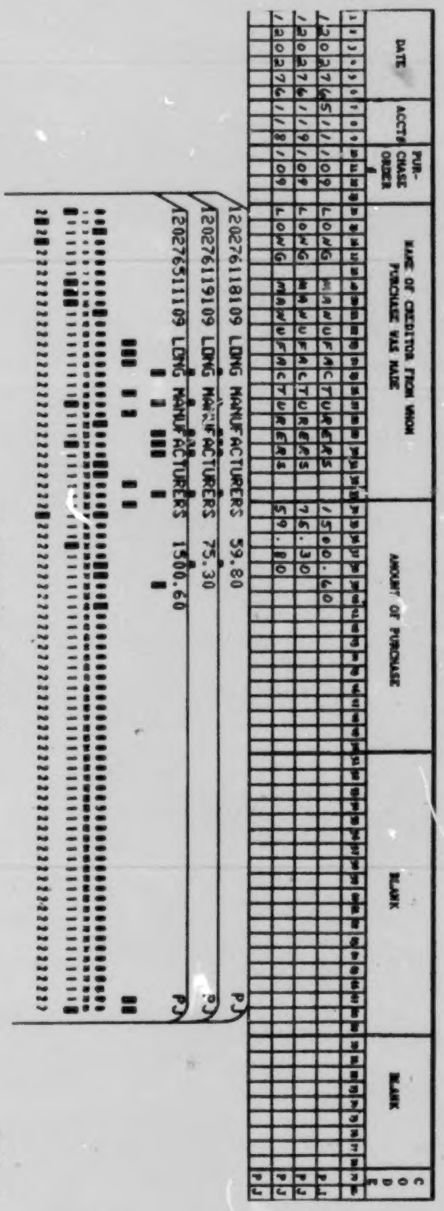
The materials developed at Danville Community College were not designed to replace the traditional accounting textbook. These materials are intended for use as a supplement to reinforce basic accounting concepts and integrate automated concepts with topics presented in the accounting textbook. The flowchart shown in Figure I illustrates the procedure a student must follow when using this learning package.

The student must first analyze the accounting transactions exactly as he would using a purely manual system. When the transaction has been analyzed, the transaction must be recorded on the proper coding form. Coding forms for all journals (sales, purchases, cash receipts, cash payments, and general) are provided for the student. Figure II shows a portion of a purchases journal coding form with a transaction previously recorded. Punched cards produced from the coded transactions and purchase journal entries generated by the transactions are also shown. All journal coding forms contain a similar format for ease of keying. Transactions recorded on the coding forms must be keyed on some input medium (such as punched cards) and submitted for execution. Keyed transactions are edited by the package for keying or deck assembly errors. Detection of any errors of this type will produce an appropriate error message. Accounting documents will not generate until the student has corrected all keying and deck assembly errors. The student must follow this same procedure when using both PHASE I and PHASE II. Portions of PHASE I output are shown in Figure III. To use the package successfully to solve accounting problems, the student must demonstrate a knowledge of accounting principles.

The package may be used to produce a completed set of accounting records (i.e. "books") for the student. PHASE I will produce all journals, the general ledger, and trial balance. Account balances from the PHASE I trial balance and adjustments serve as input to PHASE II which records adjusting entries and produces financial statements.

Advantages Derived from Integrating Computerized Concepts into Principles of Accounting

This computerized approach to the teaching of Principles of Accounting presents the following advantages to the accounting student:


PURCHASES JOURNAL OR VOUCHER REGISTER
DATE
ACCT#
PUR-CHASE ORDER #
NAME OF CREDITOR FROM WHOM PURCHASE WAS MADE
AMOUNT OF PURCHASE
BLANK
BLANK
CODE
120276 5 1 1 109 LONG MANUFACTURERS 1500.60 PJ
120276 1 1 9 109 LONG MANUFACTURERS 76.30 PJ
120276 1 1 8 109 LONG MANUFACTURERS 59.80 PJ
PJ
120276118109 LONG MANUFACTURERS 59.80 PJ
120276119109 LONG MANUFACTURERS 75.30 PJ
120276511109 LONG MANUFACTURERS 1500.60 PJ

```
                                   ***PURCHASES JOURNAL***                                        PAGE  1

                                          ACCOUNTS                 STORE      OFFICE     SUNDRY ACCOUNTS (DR)
                               ORDER      PAYABLE    PURCHASES    SUPPLIES   SUPPLIES
DATE  ACCOUNT CREDITED         NUMBER      (CR)        (DR)        (DR)       (DR)      ACCOUNT              AMOUNT
---------------------------------------------------------------------------------------------------------------------

12- 2  LONG MANUFACTURERS       109       1500.60     1500.60

12- 2  LONG MANUFACTURERS       109         75.30                            75.30

12- 2  LONG MANUFACTURERS       109         59.80                 59.80
                                          --------    --------    --------   --------                        --------

                                          1635.70     1500.60      59.80      75.30                             0.0
                                          ========    ========    ========   ========                        ========
```

Figure II

1. The student is relieved of tedious calculations - almost all arithmetic operations are performed by the computer.
2. The boredom of the posting process is eliminated.
3. The student has a complete set of books for analysis.
4. The student is introduced to accounting methodology which closely parallels actual business practice.

The output from the package can also be used to introduce the students to the concept of an audit trail since the student has a complete set of books available to him for examination. The accounting instructor can very easily formulate exercises which require the accounting student to trace the effect of various transactions from initial recording through all phases of the accounting cycle. In addition to the keying edits (error checks), various other checks for errors are made at key points in the processing of the accounting transactions. As an example, all general journal entries are checked for equality of debits and credits. Checks of this type enable the instructor to locate rapidly the particular part in the problem where the student may have erred.

The computer has without question become the tool of the accountant. It is, therefore, essential that the accounting student become familiar with this tool if he is to be successful in the business world of today.

```
                              ***CASH RECEIPTS JOURNAL***                                          PAGE  1

                                        SUNDRY    SALES    SALES    SALES    SALES    SALES  ACCOUNTS  SALES
                              RECEIPT  ACCOUNTS  DEPT. A  DEPT. B  DEPT. C  DEPT.D    TAX      REC   DISCOUNT    CASH
DATE   ACCOUNT CREDITED        NUMBER    (CR)     (CR)     (CR)     (CR)     (CR)    (CR)     (CR)     (DR)      (DR)
-------------------------------------------------------------------------------------------------------------------------

10- 2  NOTES RECEIVABLE         100     400.00                                                                 400.00

10- 2  INTEREST INCOME          101      12.00                                                                  12.00

10- 5  R. A. BARNES, INC.       102                                                         800.00    16.00    784.00

10- 6  FOGARTY AND JACOBS       103                                                         625.00    12.50    612.50

10- 7  CASH SALES               104             1700.00                             68.00                     1768.00

10-10  DAVID T. MATTOX          105                                                         600.00    12.00    588.00

10-13  STANDARD SUPPLY CO.      106                                                        1604.00    32.08   1571.92

10-14  CASH SALES               107             1632.00                             65.28                     1697.28

10-17  ADLER COMPANY            108                                                         725.00    14.50    710.50
                                       -------- -------- -------- -------- -------- -------- -------- -------- --------

                                         412.00  3332.00      0.0      0.0      0.0   133.28  4354.00    87.08  8144.20
                                       ======== ======== ======== ======== ======== ======== ======== ======== ========
```

```
ACCOUNT NAME:  SALES DEPT A                              ACCOUNT NO.  411
---------------------------------------------------------------------------
DATE       REFERENCE                     DEBIT        CREDIT       BALANCE
---------------------------------------------------------------------------

10- 1      BEGINNING BALANCE                                           0.0
10- 1      SJ                                        1520.00      -1520.00
10- 1      CR                                        3332.00      -4852.00



ACCOUNT NAME:  SALES DEPT B                              ACCOUNT NO.  412
---------------------------------------------------------------------------
DATE       REFERENCE                     DEBIT        CREDIT       BALANCE
---------------------------------------------------------------------------

10- 1      BEGINNING BALANCE                                           0.0
10- 1      SJ                                        2354.00      -2354.00



ACCOUNT NAME:  SALES DEPT C                              ACCOUNT NO.  413
---------------------------------------------------------------------------
DATE       REFERENCE                     DEBIT        CREDIT       BALANCE
---------------------------------------------------------------------------

10- 1      BEGINNING BALANCE                                           0.0
10- 1      SJ                                        1396.00      -1396.00
```

```
                        ***TRIAL BALANCE***

NO   ACCOUNT                             DEBIT             CREDIT
-------------------------------------------------------------------
111   CASH ON HAND                     5044.20
112   CASH IN BANK                     2620.24
113   NOTES RECEIVABLE                  400.00
114   ACCOUNTS RECEIVAB               10918.60
211   ACCOUNTS PAYABLE                                     340.00
218   SALES TAX PAYBL                                      790.88
311   A CAPITAL                                           7560.00
411   SALES DEPT A                                        4852.00
412   SALES DEPT B                                        2354.00
413   SALES DEPT C                                        1396.00
414   SALES DEPT D                                        3670.00
415   SALES DISCOUNT                     87.08
416   SALES RET AND ALLOWS               25.00
418   INTEREST INCOME                                       12.00
511   PURCHS                           1300.00
512   PURCHS DISCTS                                         26.00
517   ADVERTISING EXPENSE               155.00
520   STORE SUPPLIES                    100.00
542   REPAIRS AND MAINTEN               350.76
                                    ----------         ----------

                                      21000.88           21000.88
                                    ==========         ==========
```

Figure III

USNA-COMPUTER SUPPORTED INSTRUCTION SYSTEM: SOPHISTICATED, GENERATIVE AND EASY TO AUTHOR

Richard A. Pollak & James W. Schwab
Academic Computing Center
U. S. Naval Academy
Annapolis, Maryland 21402

The Naval Academy has developed a Computer Supported Instructional System (CSIS) that is sophisticated yet easy to use. The instruction sequence is authored through a prompting sequence. The questions are generated from item forms, and the answer-checking routines allow single numeric responses or whole sentences to be analyzed. The author can specify up to six predictable wrong answers and one unpredictable response in which diagnostics can be presented or branching sequences can be specified. The author is able to select from three basic learning strategies, ranging from very simple to sophisticated. Instruction has been developed in physics, economics, chemistry, and English, with navigation and electrical engineering sequences on the way.

Over the last 2½ years the United States Naval Academy has undertaken the project to develop a generalized drill/practice/tutorial Computer Supported Instructional System (CSIS).

The main goal in designing CSIS was to provide a means of computer support of course material for use in all disciplines with equal effectiveness. Another goal was to provide a system that would be easy to author yet sophisticated enough to be used effectively in advanced technical subject areas as well as in the humanities. Both of these goals have been met by the USNA CSIS system.

The person who prepares the material for the system will be referred to as the author. The author here is intended to be a subject matter expert, not a computer programmer; thus only limited knowledge of the USNA/Dartmouth Time-Sharing System is required.

The system provides feedback for both the students and their instructors. The author can specify the content of each of the following three forms of student feedback: (1) one of several congratulatory messages for a correct response, (2) an appropriate diagnostic message for up to six predictable wrong answers (PWA), and (3) an optional message for an unpredicable wrong answer. Further feedback is provided to the students at the completion of each exercise, giving them a user summary that indicates their own performance on the exercise plus an opportunity to comment on anything they wish to about the exercise and/or the overall system. Instructors receive feedback by analyzing statistic files that have been collected from the system while the student performs the exercise. These files contain such things as user performance summaries, error responses, and the student comments that are recorded at the completion of each exercise.

CSIS offers many features that make it both simple and flexible. One of the main features allows the author to build an exercise in basic building blocks called "frames." A frame may be in the form of a question, a comment, or a help sequence; or it may be a means of turning over the control of branching to the student. The frame type designator determines how the computer is to analyze the student's response, if a student response is necessary. Another important feature is that frames may contain numeric and string variables and function expressions -- all under the control of the author. This feature allows the author to present the same "item form"* to all students, but with each student getting a different set of values for the variables and functions used in the frame. Furthermore, the author can control such things as: the number of tries a student is allowed for answering a question correctly; whether or not the frame contains graphics; the number of times a frame is to be presented, substituting different variables; allowable correct answers; optional predictable wrong answers and diagnostics; and optional unpredictable diagnostic messages. The system allows the author to designate whether the student receives the exercise in drill or quiz mode. The author can also determine whether the student should be allowed to use such special features during a session as: calculator mode or selection of specific frames.

---

* An item form, which defines a class of items characterizing a task, has the following characteristics: (1) it generates items with identical structural characteristics, (2) it contains one or more variable elements, and (3) it defines a domain of items.

This system offers the author two modes of input: one is a conversational mode prompting program, and the other is a preprocessor mode for quick entry. The first prompts the author for all input necessary to build an exercise file -- and this may be the best way for novice authors to begin using the system. The latter program, more suitable for advanced users, requires the author to build a source file in a specific format that will be the input files to the preprocessor.

CSIS has been designed for its simplicity so that anyone can build introductory level CSIS exercises; and, with experience, more sophisticated exercises can be authored, using the more powerful features and branching techniques the system provides.

CSIS operates on two modes: drill and quiz. Drill mode (Drill Mode refers to Drill/Practice) takes advantage of such CSIS features as Predictable Wrong Answer (PWA) checking which helps the student by printing diagnostic messages that relate to his/her answers. Drill mode lets the student have as many attempts to respond with the correct answer expression if the student fails to respond correctly in the specified number of attempts. In quiz mode the CSIS does not do any PWA checking or carry on a dialogue with the student. It merely presents each frame, evaluates the student's response and goes on to the next frame. Both modes supply the students with a summary of their performance on the exercise.

The CSIS exercise is designed around basic building blocks called **FRAMES**. There are basically five different classes of frame types used in CSIS: Numeric, Keyword (or String), Command, Branching, and Survey. The frame type designator is used by the author to tell CSIS how it should analyze the student's responses.

♦ NUMERIC % - (Frame Type 1) - is the more sophisticated of the two numeric frame types. The student's response, a numerical value, is compared to the value computed from the author's correct-answer expression according to numerical tolerance limits based on percentage error acceptable by the author. The limits define how much error will be acceptable by the author to still consider the answer "correct", how much error would be considered a "close" answer (but evaluated "correct"). For example, the author may specify the following:

1) Within 1% - - "Correct"
2) Greater than 1% but less than 3% - - "Close"
3) Greater than 3% - - "Wrong"

When CSIS is evaluating the student response for this frame type, it first checks to see if the response has the correct number of significant figures. If not, a warning is printed. The author can suppress significant digit checking if so desired. The student response is compared to see if it is within the correct limits (the Correct-Answer value plus or minus a specified percent of the correct value). If that test fails, the response is checked for a bad decimal placement. If this fails, the response is compared against any Predictable Wrong Answer (PWA) expressions the author has supplied for the frame.

♦ NUMERIC RANGE - (Frame Type 2) - the numeric range frame type is a simpler numeric type that doesn't worry about significant figures and doesn't check decimal placement. The student response, a numerical value, is compared against a value computed from the author's correct-answer expression, using a different set of limits than the Numeric percent. CSIS compares to see if the student response is between the correct value minus a low limit and the correct value plus a high limit. The author is responsible for setting the prescribed low and high limit which may be algebraic expressions.

♦ EXACT ANSWER - (Frame Type 3) - is used by the author to make the student respond with an answer that matches exactly with one of the author's acceptable correct-answer expressions. This frame type requires exact placement of commas and spaces and will accept no more, or no less, than the author's acceptable correct answers. This type is useful for short answers, where wording and/or spelling must be precise, multiple choice, and other type objective questions. A word about keywords needs to be discussed. The advantages of using keywords is that in answer checking, the author need not be concerned with: exact spelling by student, correct word endings (e.g., -ing, -ed, -s, -es, etc.), prepositional phrases. For example a question could be:

Question: **WHERE IS THE COMPUTING CENTER LOCATED?**
Correct-Answer Expression: **WARD**

The student could respond to this question in the following manner: **THE COMPUTER IS IN WARD CENTER.**

1. In using keywords CSIS searches through the total student response looking for the letter **WARD**, the above student answer will be counted correct.

2. If the author would have designated the correct response to be: **WARD HALL** the above student response would be wrong.

3. If, on the other hand, the student responded with **THE COMPUTER IS IN WARDOW HALL** CSIS would consider this response as correct because it is looking for **WARD** and **HALL** and both these exist in some form in the sentence. The disadvantage of keywords is that the system may accept a wrong answer as correct. The author needs to express his/her answer in limiting keywords but not be too restrictive so that the advantages of keyword still exist.

♦ Keyword (OR) - (Frame Type 4) - The keyword (OR) frame type is used when one or more characters, words, or phrases can be accepted for a student response. If the student response contains any of the keywords within the Correct-Answer expression, the response will be evaluated correct. The author can provide up to 10 of these expressions for any keyword frame type.

♦ Keyword (AND) - (Frame Type 5) - In using Keyword (AND) the author again sets up the Correct-Answer expression in terms of keywords. With this frame type, CSIS will require the student response to contain all keywords listed in a given Correct-Answer expression. Order of the keywords in the student response is not important.

♦ Keyword (ORDER) - (Frame Type 6) - This frame type is set up exactly like Keyword (AND). The only difference is that each keyword or phrase in the student response must be in the same order as it appears in the Correct-Answer expression.

♦ COMMENT - (Frame Type 7) - The COMMENT frame type requires no response on the student's part. It is used simply to give the student instructions, present new material or data, or to give remedial help.

♦ BRANCHING - (Frame Type 8) - The BRANCHING frame type is used to let the students have control over their own movement through a CSIS exercise. Depending upon the student response, CSIS will branch to a designated frame and continue from there. The author may provide from two to ten branch controls for each branching frame.

♦ SURVEY - (Frame Type 9) - The SURVEY frame type for the student response is either trapped in a survey file created by the author or used as a variable, or both. This frame type can be used to record the student responses to a survey (or questionnaire type) question where there is no right or wrong answer. Another example might be to let the student enter data values for an actual laboratory situation so that CSIS can ask questions.

CSIS has built into it three basic strategy modes. They are used to control the number of times a given frame or item form is to be presented to the student, how many presentations the student must answer correctly to proceed on the success branch, and to what frame CSIS should present to the student next.

Strategy #1, the frame or item is presented to the student a specified number of times, and CSIS branches to the same next frame whether or not the student gets the answers of the individual presentations right or wrong. (Linear-non branching sequence). This is the simplest teaching strategy and can be used to present a series of questions in which all students take the same path through the instructional sequence. This strategy can also be used for comment frame types where there is no student response necessary, or for survey frame types where the student response is trapped but not evaluated as correct or incorrect.

Strategy #2, the frame or item form is presented to the student a specified number of times as in Strategy #1. Additionally, after the student has completed the total number presentations specified, CSIS then determines if the student has answered enough of the presentations right to go on to the success branch. If the student does not answer enough right, then the author can direct the student to a help sequence or to remedial work.

Strategy #3 is similar to Strategy #2, except that CSIS checks after each presentation to see if the student has enough correct responses to go on to the success branch, or enough wrong to go on to the failure branch. This way the student does not need to receive the total number of the presentations of the frame specified by the author.

In summary, the major goal in developing a generalized drill/practice/tutorial model is to design a system that can be applied in all disciplines with equal effectiveness. The objectives of CSIS are as follows:

1) The author is the subject matter expert, not a computer programmer.
2) The questions are to be generated from item forms.
3) The answering routines allow single numeric responses or a whole sentence to be analyzed.
4) The author is able to specify up to six alternative responses (predictable wrong answers) in which diagnostics can be presented to the user. The author is also able to supply a diagnostic help sequence for an unpredictable response.

5) The author is able to select from among designated learning strategies, from very simple to sophisticated with allowance for a minimum of effort and a maximum of flexibility.
6) A comprehensive history of each student is kept as he or she progresses through the system.
7) The model allows the author the capability of choosing a quiz(in which all diagnostics are suppressed) or a drill/practice or tutorial in which all the diagnostics are presented.
8) The author has the flexibility to control whether or not the student has the ability to use special functions of the CSIS such as (a) calculator mode; (b) selection of specific problems; (c) skipover questions; (d) stop program; (e) use of upper/lower case characters; (f) random congratulatory responses; (g) restart from beginning.

The system has many features that make it unique as well as flexible. Questions can be put into a file via conversational mode, or, as happens many times after a user becomes familiar with the system, by means of a short cut. A preprocessor has been designed which circumvents the conversational mode for quick entry. The conversational mode allows the computer to ask the necessary questions to build a question file. The text of the question can be variable number of lines with a maximum of 25. The question can contain: computer-generated numbers within specific parameters; words, letters, or phrases randomly picked from a category file or in a specific sequence; or a function generated from a BASIC language statement. The question-answer analyzer can analyze up to nine different types of numeric and string responses. The author has control of such things as number of attempts allowed for each question, number of question presentations using different variables for the same question. The author can choose among several different learning strategies built into the system or define new ones.

The power of this system lies in its ability (1) to generate questions within specific parameters, (2) the ability to analyze answers (numeric or sentence) that the students type in, and (3) the ease with which the user can author programs.

Curriculum is presently being developed using the USNA-CSIS in Physics, Economics, Chemistry, Electrical Engineering, Navigation and English.

Great appreciation goes to E. Carolin Stephens for all her help and endurance throughout the documentation of the CSIS system.

INTEGRATING COMPUTER USE INTO AN UNDERGRADUATE MATHEMATICS COURSE

KENNETH PRESKENIS
Department of Mathematics
Framingham State College
Framingham, Mass. 01701

ABSTRACT. This article presents some guidelines which might help a department of mathematics and/or one of its teachers in performing the activity of integrating computer use into a course.

During the fifties, Sputnik sparked the "great race" in science and technology between the U.S.A. and Russia. This race coincided with a sharp expansion in the population of undergraduate students. During this time the method of transmitting information to students, especially in undergraduate mathematics courses, was lecturing. Large numbers of students were processed by mathematics teachers through highly disciplined note-taking. This compact and concentrated educational system constrained the imagination and individuality of many students; predictably, it would have to be altered. However, the change came quicker than expected and a major force in that change was the spread of computers in society. For today's mathematics students learning about the computer has become an important part of their education and the subject matter involved is almost incompatible with the lecture method of communication. As a result, many college mathematics instructors are trying to deal with the problem of changing an established pattern of presenting material in order to accomodate the growing value of the computer as a readily available resource.

This paper speaks to the problem by presenting some guidelines which might help a department of mathematics and/or one of its teachers in performing the activity of integrating computer use into a course. The word "guidelines" should be taken literally since a precise description of this activity is impossible varying with the teacher who would perform it, with the material in the course and with the students whom the course serves. The paper draws on the author's experience over several years in both a small college and a large university (Boston College). As far as computer facilities are concerned, the large university had an IBM 370 computer with time sharing capability for a large number of terminals and the small college had merely one terminal linked over a telephone line to a Hewlett Packard 2000 computer. The guidelines offered here can be implemented using the small college's facilities with very little loss of value.

In order to put the guidelines into proper perspective some underlying assumptions need to be verbalized. First, "computer use" is taken in a very general sense referring to most appearances of a computer in an instructional setting. Second, "integrating into a course" does not involve the notion of adding new material to a course, rather it presumes the existence of a solid instructional unit and looks towards enhancing the effectiveness of that unit. Next, the guidelines presented here do not require a teacher to get deeply involved in technology. However, they do assume that a teacher is willing to work hard to diminish any personal constraints to the computer and that a teacher will work with well prepared computer programs.

A natural way for a teacher to begin integrating computer use into an undergraduate mathematics course is to confront the question - What is to be gained in trying to do it? The first response to the question is that the activity will call on and will weld together many elements of the teacher's professional side. For example, in order to become educated the teacher would establish a cadre of colleagues (in mathematics or in education) who worry about this kind of task regularly; on a broader scale the teacher would persistently build up a mental frame of reference related to the task from many sources such as written resources or short conferences at other schools. Thus, the activity will be a positive force in bringing out and in sustaining the full professionality of the teacher. Next, the activity will get the teacher involved with the subject matter of the course in such a detailed way that the actual presentation of the course can be significantly improved. For instance, there would be constant analysis of the subject matter to see whether it would accomodate a computer use and whether the combine of the two would improve the effectiveness of the teaching for the students involved. Finally, the activity of integrating computer use into a course cultivates the will and facility to improve one's teaching by causing in the teacher some real contemplation on teaching style. This reflection on style stems from decision after decision being made about the compatibility of particular computer uses with the teacher's manner of presentation.

The first step in the activity of integrating computer use into a course is the orientation of both department and teacher to the task. The departmental view should be that of a legitimate professional activity which contributes directly to the teaching of mathematics (thus, indirectly to the field of mathematics) and which results in better service to students in the broadest sense of these terms. When a teacher is trying to integrate computer use into a course, the department should see one of its members trying to use a unique resource to arrive at a "better" sequencing of mathematical material and at a "better" use of student's (mathematical) time. Having the teacher do this is valuable for the department as a whole not only in the abstract sense as a manifestation of the department providing an environment most conducive to the enlargement of its members but also in the practical sense as an experiment whose results may very well have ramifications for the departmental program and future students. On the teacher's part, integrating computer use into a course should be seen in the larger context of the departmental program and should draw on carefully thought out departmental ideas about the course's role in the program as a whole. Ideally, the teacher should have typical academic freedom in decision-making about the course and should operate in an environment where it is legitimate to fail as well as succeed. On a human level the teacher should feel the presence of a perceptive and supportive (but non-imposing) department. Concerning the computer itself, the teacher's perspective should be one of improving the course through the use of a new and unique resource, perhaps, analogous to drawing from an established pool of guest speakers. For instance, suppose the computer has an excellent random number generator; the teacher should not only screen different parts of the course to see if they are adaptable to randomization but also, having found one part, should rethink that part to determine whether the computer use will actually aid the presentation.

After orientation, one of the most important steps for the teacher is the internalization of a clear concept of the course and its purpose. There is almost no way to overstate the importance of this step since this will guide the teacher in both organizing the topics and selecting the instructional procedures to convey those topics. In particular, computer use will be thought about in relation to something abstract outside the course itself. In a sense, the day to day running of the course will be the externalization of the concept (and purpose) of the course. In order to illustrate this important step, the author would like to draw on two personal experiences of integrating computer use into a course, namely, a differential calculus course for pre-medical students and a one semester mathematics course for humanities students both taught at Boston College (in 1975 and 1976 respectively). For the calculus course, the author was inspired by a book entitled "Functions and Graphs" by I.M. Gelfand, E.G. Glagoleva and E.E. Shnol (see (2)) and a computer program available from Hewlett-Packard (see (4)). These led to an initial concept of the course as a course in graphing in the cartesian plane. The initial concept gradually developed into a single line of thought with a single point of view i.e., each part of the course would concern itself with graphing and interpreting ideas geometrically; further, this concept would become quite clear to the students. Next, by trying to understand and appreciate what would be most beneficial for the students involved, a formulation of the purpose of the course was reached, namely, to convey the three ideas of function, derivative, and the interplay between the two plus some pertinent skills and language to students who will have to bring calculus to bear in their scientific work. The formulation of this purpose led to the notion that the students should not only be able to perform conventional operations in calculus but also be able to conceptualize so well that they can move easily between ideas and applications. This notion joined with the concept of the course to result in a course which conveyed the major ideas of differential calculus using a geometric approach and which included computer programs whose function was to enhance the geometric interpretation of the ideas. For more information on this course the reader should see (1). For the Liberal Arts course the author was responding to a need of the so called Honors Program for a new mathematics course which would service students whose overall high school performances were above average but whose mathematics performances were average or below. The initial concept of the course was that of an introductory course in mathematical and computer modeling. The concept eventually became that of a course which would use carefully selected mathematical models as a vehicle in the exposition of certain major ideas in mathematics so that the students would see the usefulness of what they were learning. The purpose of this course was the development in the student of some understanding of and some real altitude towards mathematical ideas (with little emphasis on skills). Of course, this development was constrained in a natural way because of the students' limited disposition and capacity for mathematics. However, there was a belief that most thinking people deal with problems in the real world by generating a mental model of those problems and to the extent that this belief was true for the students concerned, the idea of generating a non-mathematical abstract model to deal with a problem played a helpful role as a starting point and constant reference point throughout the course. Furthermore, the computer operating in its algorithmic request-response way and supplying a user generated output sheet played an important role in "slowing down" the mathematical analysis of the models and in helping students make and confirm judgements. For more information on this course the reader should see (3).

After getting oriented and settling on a concept (and purpose) of the course, the teacher is ready for the delicate professional job of translating this concept into an organization of topics and instructional procedures which include computer uses. Initially the teacher should analyze the subject matter to separate the main ideas from the minor ones. Once the main ideas are identified, they should be treated as the real substance which is to be transmitted to the students. The next step is for the teacher to design a process aimed at conveying the substance to get at the objective of the purpose of the course. (The objective will usually be a certain development inside the student.)

The teacher should temporarily discard any usual sequence of topics presenting the main ideas and try to discover an optimal sequence; optimal, in the sense of representing the concept and serving the purpose. Next the teacher should compartmentalize the sequence into small so called modules which can capture the interest of the students or which can convince the students that what they are learning is useful. Each of these modules should cover a well determined amount of material in a consistent way over an average time period of about three weeks. The different modules should be screened as much as possible to see if they can benefit from any of the unique aspects of the computer. Once a module is selected, the teacher should review and rethink its material trying to weave the computer utilization into it. It is at this point that the teacher must be willing to shed the role of direct dispenser of knowledge and to let the computer play that role. Only when a computer use has blended in with the larger instructional unit - a module - is it integrated into a course.

In the actual implementation stage of teaching a computer-integrated course, individual teaching ability and style take center stage. Hence, it is hard to say things that have general or uniform application. However, there are some ingredients which seem to be important for the success of such a venture. First, there should be a strong belief by the teacher that the course with its concept, purpose, organization of topics, and instructional procedures will benefit all concerned and also a disposition to carry this belief to all parts of the course. There should be an adoption of an atmosphere of experimentation by both teacher and students about the computer uses. The teacher should moderate the commitment to any particular computer use by having an open flexible attitude toward the details of the computer use. In other words, the teacher's involvement when using the computer should be deep but still non-intense. Finally, at the heart of the instruction is the teacher and students interacting in some context while dealing with mathematics. For each student, there will be a (relatively) fixed number of these interactions; the ultimate goal of the teacher being to make as many of these as possible be beneficial to the student. When an interaction involves the student and a computer or a computer program (not the teacher directly), there is a real problem of depersonalization in instruction and some compensation must be made on the part of the teacher. Perhaps, the teacher should carefully trace the student's path through the computer use remembering that the student being a novice will be quite suprised each time the computer does not work. At the very least, complete and detailed directions about the computer use should be given to the student. Further it is strongly recommended that genuine and sincere verbal encouragement be built into any computer aided instructional unit.

Bibliography

(1) "Computer Assisted Instruction in Differential Calculus", K.J. Preskenis, (Proceedings of 1976 Conference on Computers in Undergraduate Curriculum. SUNY at Binghamton; pgs. 229-232)

(2) "Functions and Graphs", I.M. Gelfand, E.G. Glagoleva, E.E. Shnol, (A book - from a series called Library of School Mathematics - translation and publication by the M.I.T. Press)

(3) "Interweaving Mathematics and Computers for Liberal Arts Students", (Proceedings of 1978 Conference on Computers in Undergraduate Curriculum. Denver, Colorado)

(4) "Project SOLO Computer Topics-Mathematics Projects", (A publication based on materials produced in a joint effort of the University of Pittsburgh and the Pittsburgh Public School System under the directorship of Dr. Thomas A. Dwyer with copyright by Hewlett Packard Company of Cupertino, Ca.)

INTEGRATING COMPUTER BASED SURVEY ANALYSIS INTO UNDERGRADUATE SOCIAL SCIENCE COURSES

Charles L. Prysby

Department of Political Science
University of North Carolina
at Greensboro

This paper contains a report of an evaluational project that examined five different computer-based instructional packages in three different course contexts. The general conclusion of the paper is that computer-based survey analysis is a particularly practical and worthwhile way of introducing undergraduate social science students to empirical data analysis. Some specific suggestions are made concerning the appropriate strategies for using such materials.

Perhaps in every discipline there is a tremendous gap between what is being done and what is being taught. Certainly, such a gap exists between the research activities of social scientists and their teaching activities in undergraduate courses. Considerable research emphasis is placed on formulating and testing empirical relationships, but undergraduate students generally are expected to simply read about the approaches and findings of others. Although a substantial portion of scholarly publications have a quantitative component, undergraduate students are usually asked to produce papers containing description, speculation, and unsystematic analysis. And while most social scientists see systematic empirical research as a fundamental part of their discipline, they frequently fail to effectively communicate this to their undergraduate students.

To be sure, this overstates the case. Many undergraduate courses do require students to collect and analyze data, and most departments offer at least one course specifically on research methods. Nevertheless, such innovations in the undergraduate social science curriculum seem confined primarily to more advanced courses, with systematic empirical investigation remaining largely absent from most lower-level courses.

This lack of student involvement in empirical investigation is not a desirable state of affairs. It is likely to result in students who have underdeveloped analytical skills, little appreciation of the advantages and the limitations of empirical research, and a poor understanding of the nature of social science. Even in introductory courses, where considerable emphasis must be placed on communicating basic information to students, some introduction to empirical research can be offered. While this might be done in a variety of ways, having students become actively involved in empirical investigation is likely to be a very effective means. Requiring students to formulate testable statements, work with a set of data, and interpret their empirical findings should produce a deeper understanding of empirical analysis than a passive reading of materials or listening to lectures.

Some may feel that it is impractical to have students in lower-level social science courses engage in empirical analysis, either because the students lack the necessary statistical background or because the appropriate materials are not available. Neither of these reasons is a valid objection, however. Students can be taught to read and interpret contingency tables, allowing them to work with survey data. Also, relevant materials are available. For example, computer-based instructional packages that couple a data set to a statistical routine have been developed, and many of these are appropriate for introductory-level courses. Five such instructional packages are discussed in this paper. Moreover, an interested instructor could develop appropriate materials--perhaps by drawing a subset of variables from an existing study, placing these data on a computer tape or disk, and preparing a mimeographed handout containing user instructions.

This paper attempts to contribute to our understanding of how computer-based data analysis can best be used in lower-level social science courses. Although it is not necessary to use the computer in order to have students analyze data, computer-based materials are probably more efficient for this purpose. Such materials minimize the tedious and boring aspects of data analysis and maximize the amount and flexibility of data that can be made available to students. For these reasons, the focus is solely on computer-based data analysis, although some of what is contained here may apply to analysis by other means.

Essentially, this paper contains (a) a report of an evaluational project I conducted, and (b) observations and conclusions based on that experience. The evaluational project was designed to assess the value of including computer-based materials in introductory-level political science courses. The students involved in this project were undergraduates with average academic abilities and little exposure to political science or the computer. Most of these students were unaware that computer-based materials were to be used when they enrolled in the course. Hopefully, the parameters of this evaluational study permit generalization to a variety of basic social science courses at a variety of institutions.

Materials and Setting

In order to evaluate the usefulness of computer-based materials, five different instructional packages and three different courses were involved in this project. The first stage of the project involved integrating one instructional package into a regularly offered introductory political science course. This was part of a larger project being carried out by CONDUIT to evaluate the transportability and usage of computer-based instructional packages designed for undergraduate social science courses. Following this stage, two other instructional packages were used in an introductory public opinion course, a fourth package was used in a voting behavior course (offered at a somewhat more advanced level), and a fifth package was introduced into the course first mentioned. In most cases, the usage was repeated in subsequent semesters. Space limitations prohibit a detailed discussion of each attempt, so we shall concentrate on the first stage of the project. The following stages basically were variations on the first attempt, and they shall be summarized very briefly.

Two sections of an introductory-level course, Democratic Political Systems, were involved in the first portion of this project. Each section contained about 30 students, primarily sophomores and juniors without previous exposure to political science or the computer (70% had not taken a previous political science course; 95% had never used the computer in any previous course). The instructional package used (Boynton, 1972a) was selected because the data set includes survey data on four Western societies (United States, Great Britain, West Germany, and Italy), all of which were covered by the course readings. The accompanying student manual includes user instructions, analysis projects, and a codebook for the data set.

The remaining stages of this evaluational project involved using similar materials, with variations in the extent, strategy, and context of usage. One attempt involved extensive use of computer-based materials in a course on public opinion, composed almost entirely of freshmen and sophomores without previous exposure to political science or the computer. Two different instructional packages were selected for use: one package (Boynton, 1972b) contained data drawn from five presidential election surveys, coupled with a manual similar to the one described above; the other instructional package, adapted from Dartmouth's project IMPRESS (see Denk and Moxley, 1972), contained variables drawn from a 1968 election survey. Other efforts involved using materials from the SETUPS series sponsored by the American Political Science Association. Two different packages (Arterton and Hahn, 1975; Bowen, et al., 1975) were used in two different courses. Both of these computer-based instructional packages have a data set that is drawn from an existing national survey and an instructional manual that includes a codebook, user instructions, and relevant textual material.

The ability to use the above materials obviously depended upon having the necessary facilities available. In this particular case, the university benefits from its membership in a regional computer network, which gives users access to an IBM 370/165, to a data library, and to support staff at the regional computer center. Linkage to the IBM 370/165 was through a UCC 1225 medium-speed terminal. All use of the materials by the students was through remote batch processing procedures, with jobs being run overnight and the output returned the next day. The Statistical Package for the Social Sciences was used to analyze the data for four of the packages (with the data stored as SPSS files); one package (Denk and Moxley, 1972) contained its own analysis program.

Objectives and Strategies

What an instructor chooses to do in a course obviously depends upon what he wants to accomplish. Thus, some mention of course objectives should precede a discussion of educational strategies. In both courses involved in this project, the instructional use of computer-based materials was expected to both (a) develop analytical skills and a basic awareness of empirical investigation and (b) develop substantive understanding of the course topics, or at least some of them. The first goal included having students learn (1) how to formulate and test relationships between variables; (2) why such relationships are of particular interest to social scientists; and (3) what problems are involved in the use of empirical data to examine political behavior. The second goal involved (1) providing basic information about the subject matter; (2) introducing key concepts and approaches; and (3) stimulating thought about the implications of the findings.

In the first portion of this project, involving the Democratic Political Systems course, the computer-based materials were not used until the final one-third of the course. During the first two-thirds of the course, students received considerable information about the political systems of the specific countries and some introduction to relationships and contingency tables. Before submitting their first computer run, students received two lectures that described the survey study, outlined the instructional package, and explained the mechanics of making the computer runs. Handouts containing instructions on the punching and submitting of the job deck were provided to each student in order to minimize difficulties in the mechanical aspects of analyzing the data. Students had to do the analysis projects in the manual, these projects progressing from highly structured at the beginning to fairly flexible at the end. Class sessions were devoted to discussing various aspects of the analysis projects, including everything from how to interpret specific tables to what the general implications of the findings were. After completing the analysis projects, students were required to carry out an assignment that involved formulating a set of propositions, testing those propositions with the available survey data, and submitting a paper based on that proposition formulation and testing.

Similar strategies were followed in the other courses with the other instructional packages. The major points of variation were: (a) the amount of time devoted to these materials, which ranged between 30% and 60% of the total course time; (b) the degree of flexibility in the analysis assignments; and (c) the extent to which the analysis assignments were integrated with a set of substantive readings.

Results

The evaluation of the computer-based materials was based on three sources of information: (1) questionnaires filled out by the students on an anonymous basis at the end of the course; (2) written assignments submitted by the students as part of the course requirements; and (3) feedback obtained from class discussions or individual consultations with students. The evaluation of these materials involved determining: (a) the extent of difficulty students had in using the materials; (b) the degree to which the materials developed the analytical skills of students; (c) the substantive impact of the materials; and (d) the overall reaction of the students to the use of such materials.

In general, students appeared to have few problems in the technical aspects of using the packages and analyzing the data. Even for the Democratic Political Systems course, which had about sixty students altogether, I had to spend very little time helping students to get their jobs to run correctly. In fact, about 80% were successful on their first attempt, and few of the remainder had any difficulty beyond their second attempt. Furthermore, the students reported little difficulty in using these materials when filling out the questionnaires, which consisted of closed items, each with a five-point scale. Responses to the questions indicate that most students found the computer-based materials easy to use, even though many did not anticipate such ease. Specifically, 55% admitted that they were initially apprehensive of the technical difficulties of using the computer, but only 15% indicated that they had great difficulty in adjusting to the materials, and only 21% claimed that the use of such materials greatly increased their workload for the course. Comparable results were obtained from questionnaires administered at the end of the other courses involved in the evaluational project.

More difficulty was encountered in trying to develop analytical skills. A great deal of emphasis had to be placed on the fundamentals of empirical analysis, as most of the students were unfamiliar with the basic aspects of proposition formulation and testing, and many initially had problems in properly interpreting contingency tables. Despite all these difficulties, most students appeared to greatly improve their understanding of empirical analysis. There was considerable variation in the quality of the papers submitted, but most demonstrated a sound grasp of the basics of empirical investigation. Feedback obtained from class discussions and individual consultations also indicated substantial improvement in this area. While the evaluational questionnaires did not contain items that systematically and directly tapped student perceptions of how the computer-based materials aided their understanding of empirical investigation in political science, the limited indirect evidence that could be derived from the questionnaires did seem positive. For example, over 70% of the Democratic Political Systems class agreed that the concepts they learned through using the computer-based materials made them more interested in the subject matter.

It is difficult to form any conclusions about the substantive impact of these computer-based materials. While the class discussions and written assignments suggested that the students were receiving adequate information on the topics, there is no way to judge whether this was more or less information than they would have received through other means. Perhaps the important point to be made is that use of these computer-based materials, which are oriented toward the development of analytical skills, does not appear to significantly hamper the acquisition of substantive information.

The overall student reaction to the computer-based materials was quite positive. Students in the Democratic Political Systems course responded to a number of general items in a very positive fashion. For example, 77% agreed that the computer materials were a valuable addition to the course and 64% found the computer materials superior for some portions of the course. Those not agreeing primarily selected a neutral response. Only between 10% and 15% of the students appear to have disliked using the computer-based materials. Similar findings were obtained from the questionnaires administered at the end of the other courses.

In sum, all three methods of evaluation suggest that these computer-based materials were valuable additions to these two courses. Student reactions, as measured by the questionnaires, indicate that most found the materials both interesting and enjoyable. In addition to receiving a positive reaction from the students, the materials appear to be educationally worthwhile, as the written assignments demonstrated that these materials had a substantial impact on the development of analytical skills and understanding of empirical analysis.

Conclusions

Based on the results of this evaluational project, some general observations on the use of computer-based instructional materials can be made. Although subjective in nature, these observations are put forth, if for no other reason than to stimulate some thought on these points. These observations are directed specifically to the use of computer-based data analysis in introductory-level social science courses, but many of the points naturally have a much more general applicability.

At least three basic strategies for using computer-based data sets in undergraduate courses can be suggested: (a) the computer-based materials could be used on an intermittent basis throughout the course, with one or two data analysis exercises scheduled for each course topic; (b) the computer-based materials could be used during one or two portions of the course in a fairly concentrated manner; (c) one or more computer-based data sets simply could be made available for students to utilize as resources for a research paper.

The third possibility can be dismissed as a viable strategy for introductory-level courses. Students in such courses undoubtedly would lack the ability to carry out computer-based data analysis in such an unstructured environment. Even if the instructor were willing to spend a great deal of time consulting with each student about the research paper, many students would still have difficulty working on such an independent basis. This method might be appropriate for some advanced undergraduate courses, but it is certainly an unrealistic alternative for lower-level courses.

The first possible strategy listed above would be a feasible way of using computer-based materials in a lower-level course. However, this is not likely to be the most effective strategy. If just one or two analysis exercises are to be scheduled for a course topic, then each exercise would have to be a fairly simple and highly structured analysis that merely illustrates some point made in the readings or lectures. Such a set of pre-planned data analysis exercises is not likely to have much of an impact on the analytical skills or the substantive knowledge of students. The limited data analysis that could be required for each exercise would not be very challenging, and the exercises could easily become boring to students after a while. Also, the small amount of substantive information that could be present in an exercise would not contribute very much to an understanding of the subject matter, and students might even find the exercises trivial and obvious if the analysis was too repetitious of points made in the readings and lectures.

Integrating computer-based data analysis into lower-level social science courses is likely to be most successful when fairly concentrated work with the materials is scheduled for one or two portions of the course. Use of the materials might begin with some highly structured analysis exercises, to insure that students quickly learn how to submit the computer runs and interpret the output, and then progress to more flexible analysis assignments. These assignments might allow students to choose one of a set of highly related topics and carry out an empirical investigation of that topic using the available data set. Such an assignment structure would give students some freedom to choose a topic of particular interest to them, while at the same time supplying a common focus to these research activities. Providing flexibility to the assignments would be particularly advisable for introductory courses, where the students differ markedly in their abilities and backgrounds, as it would allow the more capable students to carry out more sophisticated analyses while others remain at a more elementary level.

Having students work on related but distinct topics should provide an environment that encourages meaningful class discussion, as each student would have a relevant set of findings to report and discuss. This should help generate student interest and give the computer-based materials a substantive impact. Maximization of the substantive impact of the materials also hinges on the connection between the analysis assignments and the course readings. It is desirable for the assignments to compliment but not repeat the material contained . One way of doing this would be to select readings that present general theoretical material and to construct assignments that require application of the material. What should be avoided are readings containing empirical findings that nearly duplicate the results of the required analysis assignments, as the students would receive little additional substantive information from their analysis. If the readings and assignments are well matched, and if sufficient student interest is generated, the use of computer-based data analysis should be an effective method for developing substantive knowledge of the topic.

Additionally, in-depth analysis of a specific topic should help students acquire analytical and methodological skills, as they would be required to face the problems involved in testing a set of propositions and drawing general conclusions from specific empirical findings. But in order for this activity to have the desired impact, a great deal of emphasis must be placed on the fundamentals of empirical research. For one thing, students in introductory courses frequently display a certain awe of the computer and a native faith in the accuracy of the data, which leads to the belief that the findings perfectly represent reality. In order to avoid this unquestioning acceptance of the data, the limitations of empirical research and the problems involved in generalizing from the data must be communicated to the students.

While considerable emphasis should be placed on the fundamentals of empirical research, it is important to restrict to course goals to just that. Most students will come in to introductory political science courses with very little knowledge of the basic aspects of empirical research, and it will require some time to develop these analytical skills. It is unrealistic to expect students in such courses to acquire a great deal of methodological knowledge, and it probably is best to avoid all but the most elementary statistics.

Finally, some practical considerations should be mentioned, as such factors also contribute to a successful use of computer-based materials. First of all, if the strategy recommended above is adopted, a large and varied data set should be available for students to analyze. A narrow selection of variables restricts the ability of students to choose analysis topics and a small number of respondents limits the ability to make reliable inferences about population subgroups. Both of these factors would destroy much of the advantage of having flexible assignments, and a small data set could prove to be very boring for many students. Also, many students will require individual help with their analysis, and they must have sufficient opportunity to confer with the instructor. This instructor-student interaction seems so important, at least for the strategy recommended above, that it may not be practical to integrate computer-based data analysis into very large classes, unless they are divided into smaller recitation sections capable of providing the support needed for a successful use of these materials.

REFERENCES

Arterton, C. and H. Hahn (1975) Political Participation. Washington: American Political Science Association.

Bowen, B., C.A. Broh, and C. Prysby (1975) Voting Behavior: The 1972 Election. Washington: American Political Science Association.

Boynton, G.R. (1972a) Citizens and the Political System. New York: Harper & Row.

______ (1972b) Voting Behavior in the United States: 1952-1968. Iowa City, Iowa: Laboratory for Political Research.

Denk, J.R. and N. Moxley (1972) The POISSON Primer. Research Triangle Park, North Carolina: North Carolina Educational Computing Service.

Author's Note: I wish to express my appreciation to CONDUIT (Computers at North Carolina, Oregon, Dartmouth, and the Universities of Iowa and Texas), which assisted with the evaluational project by providing a grant and one of the evaluational instruments, and to the American Political Science Association, whose Division of Educational Affairs supported my participation in the SETUPS (Supplementary Empirical Teaching Units in Political Science) project. Also, I am grateful to Ms. Maggie Davis for her assistance in the preparation of this manuscript.

224

A COMPUTER-BASED QUIZZING PROGRAM
AND ITS USE IN
TEACHING BIOLOGICAL SUBJECTS

Larry T. Spencer
Plymouth State College
Plymouth, NH 03264

In the fall of 1976, a computer-based quizzing system was instituted as an auxiliary teaching device in Introduction to Marine Life, a one and one-half credit introductory non-major science course taught by the author at Plymouth State College, Plymouth, NH. The 45 students enrolled in the course were given weekly computer generated quizzes on materials that appeared in their class text. A total of eight quizzes were given. Students were required to pass at least six. Records indicate that students required fewer attempts to pass a quiz as the course progressed. A final examination consisting of equal number of questions drawn at random from the quiz data base and from previous class examinations was given to compare student performance. Results were mixed. The computer-based quizzing approach forced students to read the class textbook more closely and at a steadier pace. It also facilitated periodic review by the students themselves with the system and the instructor in the classroom.

INTRODUCTION

Use of the digital computer for testing as well as for record keeping purposes is not extremely innovative. Infact, Franklin and Marasco (1977, p. 15) have said that "some 'interactive' computer-based testing has been no more than unimaginative implementation on computers of the most mechanical techniques of traditional testing." Yet, with large classes and a limited amount of time for giving and grading quizzes, I have used this method successfully as an auxiliary teaching device in non-major classes in the Natural Science Department at Plymouth State College, Plymouth, NH

One source of inspiration for this project came from attendance at the Third Conference on Instructional Computing held at Dartmouth College, Hanover, NH in March, 1976. One of the presenters, Dr. Allan G. Reynolds, gave a paper (Reynolds and Flagg, 1976) that dealt with his use of the computer in teaching psychological statistics. The testing system, as he described it, gave problems, accepted and graded answers, and maintained a record of student attempts and also student queries for assistance. With his system as a model, I returned to my home campus and proceeded to design and implement a similar testing system.

Plymouth State College is a small state-assisted four year college of about 2,600 FTE students located in central New Hampshire. At the time of this study all students were required to take 15 credits in science/mathematics to fulfill college wide general education requirements. Past studies by a departmental committee (Spencer, Sylvestre and Bilheimer, 1977) had showed that most students took about 12 of the 15 credits in science. The same study showed that non-major enrollments in departmental courses accounted for approximately 70-80% of the credits generated by the department. Therefore, the department has always exhibited a strong concern over non-major undergraduate science education.

The Natural Science Department offers a variety of courses in both the physical and biological sciences that students may select towards fulfilling their general education requirement. Beginning in 1973 the department departed from the somewhat standard pattern of offering three credit non-major courses by introducing the science topic concept. Science topics are one and one-half credit courses that run for eight weeks. The topics are meant to be introductory in nature, but yet are designed to cover in detail a very specific area. Subject matter covers all of science, but because of faculty availability, they have been heavily oriented towards the biological sciences. Popular topics in the past have included Endangered Species, Introduction to Marine Life, Plants of the World, Criminological Science, The Science of Hi-Fi, and Solar Energy. Enrollments in the topics are large, usually between 40 to 60 students per section. Because of the large size of the sections, most instructors use a lecture approach with multiple choice tests rather than written tests as the evaluative tool.

The goal of the department has been to teach about 23 different topics each year. Normally a few new topics are added each year and a few old ones are dropped or modified. Topics now constitute about 40% of the total credits generated by the department.

I was one of the initial supporters of the topic approach, but had long felt that one or more multiple choice lecture tests in eight weeks encouraged rote memorization and beginners luck rather than true comprehension. To promote student learning of subject matter in other classes, I normally give weekly or biweekly quizzes. With 50 students per class and only eight weeks of instruction, I did not want to waste valuable class time for that purpose in the topic courses. Thus when the school traded a card oriented batch process IBM System Three for a terminal oriented time share PDP 11/50 I quickly decided to use the computer as an adjunct teaching tool in a few of the science topics I taught. This approach seems to be very common. Hoye and Wang's 1973 compilation of programs in the area of computer-based learning lists three cases where the teaching strategy employed was a computer quiz and 27 cases where the strategy was a test.

THE PROJECT

The initial topic I selected was Introduction to Marine Life. One reason for choosing this topic was that I had taught it at least once a year since the introduction of the topic concept in 1973. Therefore, I had some previous knowledge of the levels of student participation and achievement. During the fall of 1976 when I ran the experiment, I had a class enrollment of 45 students. A second reason for choosing the course was that the text I had chosen lent itself very well to the effort. Biology of Marine Life by James Sumich (1976) has nine chapters, but in as much as the last chapter, Food from the Sea, is covered in another class, I would only need to cover one chapter per week. The book is extremely well written and has a good instructor's manual to accompany it.

At the beginning of the course the students were told that they would be participants in a project in computer-based testing. For the project, the students were informed that they would be evaluated on the following aspects:

1. successful completion of at least six of the eight quizzes given on the computer
2. level of achievement on a multiple choice final examination
3. completion of a satisfactory short paper on the natural history of a marine organism

Each quiz they took would consist of five questions and a passing score was three out of the five questions answered correctly. Students were told that they could take a chapter quiz once a day until either they had passed it or until a new chapter quiz was put on the system.

All questions for the quizzes were taken directly from the instructor's manual that accompanied the text and thus they dealt directly with information in the text rather than supportive material given in the lecture. Students were asked to take the quizzes on their own time and without looking up answers in their textbook or comparing answers with other students while at the computer terminal. Unfortunately there was no way to control access to the terminals and thus to the quizzing program. Therefore, I assume that answers were shared by many students as they attempted to pass the quizzes.

At the end of each week of class I examined the item analysis maintained by the quizzing system to determine which questions were missed by a majority of the students. During the next class session I went over the concepts involved in those questions. Although I did not specifically answer the questions I did often mention why some of the responses were incorrect.

THE QUIZZING SYSTEM

The quizzing system I designed and implemented in Basic + consisted of the following components:

1. Quiz Output Program
2. Class Record Program
3. Item Analysis Access Program
4. Various Files:
    a. Question/Response File
    b. Student Name and Achievement File
    c. Item Response File

The Quiz Output Program was accessible to all students enrolled in the course. To take a quiz the student had to enter both last name/first initial and a special identification number (given out the first week of class). If either item was entered incorrectly the student was rejected and told to re-enter the data items. If entered correctly the program then checked to see if the student had taken the quiz already that day. If so, they were told to return the following day. If at this point everything was still ok, the program told the student how many times they had taken a quiz on that chapter and what their best score was. The student was then presented with the first question/response set. The questions were chosen randomly from the question/response file for that chapter. Each time a question/response set was output ed at the terminal an internal clock was started and then stopped when the student had input ed their answer. If the time interval between question and answer was too great, the student was told that their answer would not count, even if they had chosen the correct answer, because they had taken too long. This was done to prevent students from looking up answers in their textbook. After the student answer was compared by the quiz program to the correct answer in the file, the student was told whether their response was correct or not. If an incorrect answer had been entered, the page of pages of the text where the appropriate information might be found was output ed. After the fifth question was answered, the student was told how many questions they had answered correctly and whether they had passed the quiz or not.

The Class Record Program was accessible only to the instructor. It allowed me to enter and to correct student names and identification numbers and to examine student achievement (eg., the number of times each chapter quiz had been taken and the highest score on each chapter quiz by that student) and to correct the file if the machine had malfunctioned while the student was taking a quiz.

The Item Analysis Access Program was also only accessible to me. It allowed me to examine the item analysis file maintained for each chapter. This file told me how many times a particular question had been asked and how often each of the different responses for that question had been chosen.

The question/response file consisted of questions taken directly from the instructor's manual. In a few instances I had to modify the questions to the format used by the Quiz Output Program. I also had to make up additional questions where the textbook author had been too skimpy in his instructor's manual. Each chapter question/response file consisted of from 15 to 25 questions, each of which usually had either four or five responses. In as much as the questions were drawn by the Quiz Output Program in a random fashion, all questions were used about an equal number of times.

RESULTS

Of the eight quizzes given, due to malfunctions the statistical information on quizzes four and eight was lost. Therefore the following analysis is based on six quizzes. Of the 45 students enrolled an average of 39 students took each of the quizzes. The number of times a quiz was taken varied from a high of 3.53 for quiz two and a low of 2.20 for quiz six, with the trend being in the direction of taking quizzes on a particular chapter fewer times as the course progressed. Of the six quizzes, students took on an average, 5.36 quizzes. The number of times a student had to take a particular chapter quiz in order to achieve a passing score varied from a low of 1.33 times to a high of 5.8 times, with an overall average of 2.76 times (this of course assumes that students stopped taking quizzes once they reached a passing score).

To partly assess the effect of the quizzing program a final examination consisting of 35 questions randomly drawn from the question/response file and an equal number drawn from tests used in previous years was constructed and given. The average score on this 70 question test was 56.1% (sd 12.8). When the test results were examined in terms of the percentage of correct answers to questions from the question/response file as compared to questions from past years, the scores were 66.9%(sd 17.8) and 46.0% (sd 17.1) respectively. A t-test (t = 5.0) showed that the difference between the two means was significant at the 0.001 level. Student response to the particular questions as given on the chapter quizzes and on the final showed a significant correlation (r = 0.6) at the 0.001 level. Average correct response to the questions on the quizzes was 53.9% (sd 18.8) as compared to 66.9% on the final examination. This difference was significant at the 0.01 level. A similar analysis was made of questions from previous tests. Average correct response to those items on past tests was significantly correlated to average correct response on the final examination (r = 0.7 at the 0.001 level). The total average response to the same questions on past tests was 54.8% (sd 18.7) as compared to 46.0% on the final examination. This difference is not significant at the 0.05 level (t = 2.0)

DISCUSSION

As might be expected in any kind of exercise that is associated with the computer, a number of grinches appeared on the scene during the running of the project. Due to the structure of the Basic + program, when two or more students were taking a quiz, they also accessed the same files. This posed no problems with the question/response file as it was only opened for output, but with the student name and achievement file and with the item response file, severe difficulties occurred and one student was bumped. This problem was solved by incorporating a simple shunt system within the main program that kept the second user 'standing in line' until the first user was done with the inputing operation. Since the program only recorded student achievement after a quiz was completed, this bumping did not change the record concerned with the number of times a student took a particular quiz. It did have an effect on the number of times a particular question was asked, because that item was changed any time a question was asked, regardless of whether the asking was part of a completed sequence. This bumping error was corrected within the first couple of weeks and therefore only modifies the item analyses for those quizzes.

Student access to the Quiz Output Program was also modified by the periodic downtime on the whole computer system. This caused some difficulties because not all students had equal access to the quizzing system because their own schedules. All things considered, I was rather surprized by the patience the students exhibited towards the computer. Although I had no way of measuring a frustration quotient, and although I am sure that there were many times when this quotient must have been quite high (at least general comments from students indicated that to be true), only one person said that they would not take any quizzes given by the computer. Others found the experience not necessarily enjoyable, but challenging.

I was disappointed but somewhat prepared for the rather low score on the final examination (the class average was 56.1%). Student test results at Plymouth seem to follow a bimodal rather than unimodal distribution. The number of students who 'bombed' the examination caused the class average to be lower than if they had been left out of the pool. Another reason for the low score was the poor performance on the test questions that came from previous tests. These questions were related to the subject matter discussed in class, but may have been more difficult either because of a slightly different vocabulary or because they had been used previously in a slightly different context. Since previous success for that question set had been 54%, but was only 46% for the final examination, I assume that either the students were unable to apply the information gained during the course to a new situation or were unable to understand the slightly different language of the questions.

If only the questions from the chapter quizzes are examined, it appears that students did significantly better on them when the questions were on the final examination than on the individual quizzes (66% as compared to 53%). This observation is not totally correct as the questions were given many times during the course of the quizzes and one would expect that the students would have studied less for the quiz than for the final examination. Therefore, one would assume that the question would be guessed at a number of times when it appeared on the quiz, but that the right answer was chosen if that question appeared on the final.

Student feedback to this project was generally favorable. Although student seemed somewhat awkward during their first experience at a computer terminal, they quickly overcame this and as Franklin and Marasco (1977) stated, learning to use a terminal is a lot easier than learning to drive a car. Experience though has taught me that a good sound introduction to terminal use is absolutely essential to avoid the problem of student frustration. It is also terribly important to get all grinches out of a program before opening it to general use. I didn't realize the importance of this until I used the same quizzing system with a class of 150 students and found out that what might work for 50 did not necessarily mean it would work for three times that number.

CONCLUSIONS

The question of whether students who took Introduction to Marine Life under this program know more about marine life than students who have taken the course under a different approach is not trully resolved by this project. What is apparent is that these students had to read the text more closely and at a steadier pace in order to keep up with the class requirements. Therefore, this approach definitely compensates for the'put-it-off-for-tomorrow' syndrome that characterizes many college students. This same system, slightly modified, has since been used in the same topic a second time and in a Human Anatomy course. In all instances it has allowed me to keep track of student understanding of course material without having to use valuable class time for the giving and returning of quizzes. It has been said many times that quick return of tests and quizzes provides positive feedback for the student. There can be no quicker return than the immediate response of a computer graded quiz.

REFERENCES

1. Franklin, Stephen and Joseph Marasco. 1977. Interactive Computer-Based Testing. J. Coll. Sci. Teach. 7(9): 15-20.
2. Hoye, Robert E. and Anastasia C. Wang, eds. 1973. Index to computer-based learning. Education Technology Publications, Inc., Englewood Cliffs, N. J.
3. Reynolds, Allan G. and Paul W. Flagg. 1976. Interactive Testing in Psychological Statistics (DARTS). Paper presented at the Third Conference on Instructional Computing, Dartmouth College, March, 1976.
4. Spencer, Larry T., Mark Sylvestre and Mary Bilheimer. 1977. Committee Report on the Impact of the Natural Science Topics on the Natural Science Department, Plymouth St. College, June, 1977.
5. Sumich, James. 1976. Biology of Marine Life. W. C. Brown, Dubuque, Iowa.

COMPUTER PROJECTS IN UNDERGRADUATE CALCULUS

Richard St. Andre
and
Maurice Eggen
Central Michigan University
Mt. Pleasant, Michigan 48859

Abstract: For the past several years Central Michigan University has offered through some of its sections of Calculus I and II a computer oriented approach to undergraduate calculus. Students write several computer projects in a semester in which they do their own coding using no more knowledge of a programming language than that which is taught in class. The computer is used to reinforce the learning and strengthen the presentation of the topics. This paper will describe those projects we have found most successful. For each project we will discuss (1) the objectives of the project (what the student will gain), (2) what preparations are necessary (where in the calculus sequence does it seem to best fit) and (3) related and supplemental assignments.

Introduction: For the past several years Central Michigan University has offered through some of its sections of Calculus I and II a computer oriented approach to undergraduate calculus. Because of their high degree of success these sections have become a regular feature of the Mathematics Department. The program at Central is unique in that the computer oriented sections include all the material in the standard calculus sections plus sufficient knowledge of FORTRAN to enable students to do their own coding of computer projects. They are not computer science classes. The projects are designed to enhance and strengthen the understanding of calculus.

The facilities available for use by the special sections include a Univac 1106 time sharing system, which is the primary computer for the university community, with a 1004 on line printer. Most of the work done by the students is done in batch, thus the Univac is the primary tool. There are several demand terminals which interface with the Univac available for student use (C.R.T. and teletype) but calculus use of these is limited.

In addition to the Univac the Computer Science Department has an I.B.M. 1130 computer with a 1627 CalComp Plotter attached which calculus students may make use of during selected hours in the day. The plotter is rather small and slow but the students appreciate the neatness and clarity it gives when sketching graphs and curves. The I.B.M. 1130 also has an on line 1442 card read-punch and a 1132 on line printer so that selected jobs can be submitted in batch to the 1130 computing system. The 1130 has no time sharing capability.

We have developed computer projects for virtually all the main topics in a traditional undergraduate calculus sequence. Some projects we have found successful and which will be discussed in this paper include using the computer to assist in teaching: definitions of limits, derivatives, and definite integrals; approximation techniques such as Newton's method, and Taylor polynomial approximations; differential equations (Euler's method, Runge-Kutta and graphs of vector fields); convergence of series and the First and Second Fundamental Theorems of the Integral Calculus.

I. Projects on Limits: The projects we will discuss here fit most naturally into the first few weeks of the calculus sequence. Some of them can be used to motivate definitions of limit (rigorous and intuitive), or conversely, can be used to strengthen understanding after the student has seen the definition. Other assignments can be used to motivate various limit theorems. Familiarity with functional notation and terminology and graphing of functions is all that is needed for these assignments.

Specific Assignments:

1. The standard assignment to determine $\lim_{x \to a} f(x)$ is to input a sequence of values for x approaching a and print out the

associated sequence of functional values. The student is then asked to analyze the associated sequence and determine the limit. A particularly good limit to use is $\lim_{x\to 0} \sin x/x$ since it cannot be readily evaluated by algebraic methods. See Figure 1.

```
X=          SIN(X)/X=
3.141592    0.000000
1.570796    0.636619
0.785398    0.900316
0.392699    0.974495
0.196349    0.993586
0.098174    0.998394
0.049087    0.999598
0.024543    0.999899
0.012271    0.999974
0.006135    0.999993
0.003067    0.999998
0.001533    0.999999
0.000766    0.999999
0.000383    1.000000
```

Figure 1

2. To motivate limit theorems, Assignment I-1. can be expanded to print out associated sequences for f(x) and g(x) as well as (f+g)(x), (f·g)(x), (f/g)(x), (g∘f)(x), etc. Again, the student is asked to analyze the data and hopefully draw such conclusions as the limit of the sum equals the sum of the limits.

3. The computer can also be used to indicate one sided limits, infinite limits, limits at infinity and nonexistence of limits with assignments similar to those in I-1. and 2.

II. Projects on Derivatives: These projects may be assigned as soon as the student is familiar with the concept of a limit. They are beneficial for reinforcing the definition of the derivative as a limit and motivating differentiation rules. Alternate methods of defining the derivative can be explored. Numerical techniques include using the difference quotient (f(x+h)-f(x))/h with small values of h as a replacement for the derivative. These assignments can provide an introduction to some important techniques such as Newton's method.

Specific Assignments:

1. For a given function f and a point x, examine the values of the difference quotient for a sequence of values of h approaching zero and then estimate the value of $f'(x) = \lim_{h\to 0} (f(x+h)-f(x))/h$.

2. For a given function f, the student is asked to use the plotter to graph both f and the first difference function $D_h(x) = (f(x+h)-f(x))/h$ for small values of h. Analysis of these graphs help to visualize the derivative as a function and to anticipate rules for differentiation. This project works particularly well for sine, cosine, logarithmic, exponential, linear and quadratic functions.

3. Another assignment is to plot a given function f, the first difference $D_h(x)$, and the second difference function $DD_h(x) = (D_h(x+h)-D_h(x))/h$ on the same set of axes. Analysis should reveal, among other things, relationships between the relative maximum and minima and inflection points of f and the zeros of D_h and DD_h. The function $F(x) = .0006\ x^5 - .08\ x^3 + 1.3x$ on [-10,10] is a good example to work with. See Figure 2 (in which h = .01).

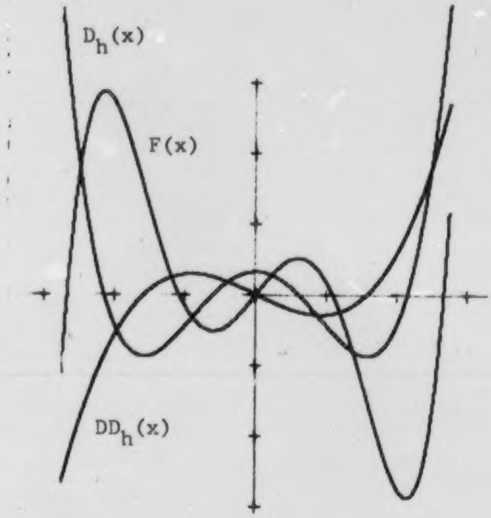


Figure 2

4. For a fixed h (say h = .001), one can print values of the difference function for various points x and functions f. The student then uses the array of data to determine some of the elementary differentiation rules. The best results are obtained with a sequence of consecutive points x = 0, 1, 2, ···, 10 and a sequence of related functions 1, x, x^2, x^3, x^4, x^5, $x^2 + x^3$, $x + x^5$, $3x^2$, ···.

5. There are several assignments that may be given related to the approximation of the zero of a function, each with its own variations. Newton's method can be easily programmed using the iterative step $x_{i+1} = x_i - (f(x_i)/D_h(x_i)$ for a small fixed h. The interval halving method can also be employed and students should compare its efficiency to Newton's method. Perhaps the most intriguing method for students is by random search: First an interval within which the function is known to have a zero is located. Then the computer is used to generate a copious quantity of random numbers within the interval. That random number with functional value closest to zero is the estimate of the zero of the function.

6. The chain rule can be dramatically visualized by plotting a composite function such as $f(x) = \sin(2x)$ and its first difference $D_h(x)$ and observing that the amplitude of D_h is twice that of f. See Figure 3.

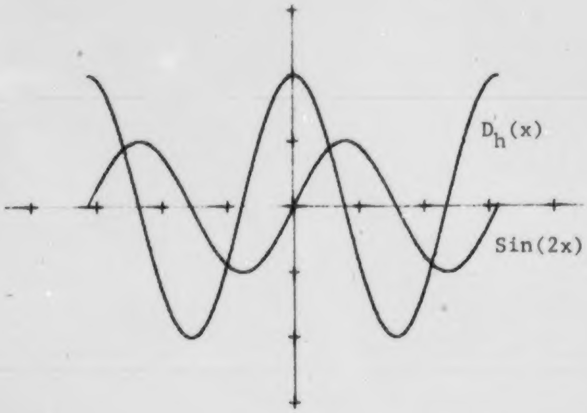


Figure 3

7. Once the definition of the derivative has been mastered, other equivalent forms may be studied via computer projects similar to II-1. Such quotients as $(f(x)-f(x-h))/h$, $(f(x+h)-f(x-h))/2h$ and $(f(t)-f(x))/(t-x)$ will appear to tend to the value of $f'(x)$ as their denominators approach zero.

8. One of the most pleasing uses of the plotter is for sketching the locus of a set of parametric equations in the plane. Students get very enthusiastic about plotting such curves as the epitrochoid, the involute of a circle, the Archimedian spiral, the Witch of Agnesi, ... We usually give this as an open ended assignment, having the students plot as many loci as their time and inclination permit.

III. Projects on Integrals: The projects discussed here should be presented in the calculus sequence at the same time as the introduction to the theory of integration. They will facilitate understanding of Riemann sums and the special limiting process involved in the definition of the definite integral. Other objectives include motivation for the First Fundamental Theorem of the Integral Calculus, the Mean Value Theorem for Integrals, and the use of approximation techniques.

Specific Assignments:

1. For a given function f on a closed interval use the computer to calculate the values of Riemann sums using left, right and midpoints of subintervals and compare the results. The student will do this for several partitions of the interval and estimate the limit of the Riemann sums.

2. A particularly simple approximation technique to program is the method of Monte-Carlo integration. Given a function $f(x) \geq 0$ on a closed interval $[a,b]$, choose a number M such that $f(x) \leq M$ for all $x \in [a,b]$. Using a random number generator produce K pairs of points inside the rectangle $[a,b] \times [0,M]$. By counting the number of points C beneath the function an approximation to $\int_a^b f(x)dx$ is $\frac{C}{K}M(b-a)$. See Figure 4.

```
 MONTE-CARLO INTEGRATION
Y = SIN(X) ON (0, 3.14)
   FOR     10 RANDOM NUMBERS,
THE AREA IS 2.199115
   FOR    100 RANDOM NUMBERS,
THE AREA IS 1.853540
   FOR   1000 RANDOM NUMBERS,
THE AREA IS 2.004336
   FOR  10000 RANDOM NUMBERS,
THE AREA IS 1.964438
```

Figure 4

3. Simpson's rule and the trapezoidal rule are easily programmed. Students can learn about the inverse trigonometric functions and about logarithmic and exponential functions by approximating the integrals $\int_a^x f(t)dt$ for appropriate f and a (for example, $\int_0^x 1/\sqrt{1-t^2}dt = \arcsin(x)$).

The students can learn about concave functions by noting that the trapezoidal approximation of an integral of a concave function is always larger than the actual value of the integral.

4. The Mean Value Theorem for Integrals can be demonstrated by partitioning the interval $[a,b]$, calculating $f(x)$ at each partition point, and averaging these functional values. For a fine partition the average value of the function is seen to be close to $\frac{1}{b-a} \int_a^b f(x)dx$.

5. A good assignment to demonstrate the Second Fundamental Theorem of Calculus is, for a given function f and a fixed h (say $h = .01$), use Simpson's rule to approximate

$G(x+h)$ and $G(x)$ where $G(x) = \int_a^x f(t)dt$, then use these values to determine the first difference for G ($D_h(x)$ as in II-2), and finally compare $D_h(x)$ with $f(x)$.

IV. Projects on Differential Equations: This collection of projects best fits into the calculus sequence after the theory of integration and a unit on applications of the definite integral. Projects in this section are designed to reinforce the notion of the definite integral, help in the understanding of first order initial value ordinary differential equations and provide exercises in the use of numerical approximations.

Specific Assignments:

1. For a given initial value problem $y' = f(x,y)$, $y(x_0) = y_0$, examine the approximate solutions $y = y(x)$ as provided by Euler's method, the improved Euler method and the Runge-Kutta method for various step sizes h. For those differential equations for which a closed form solution exists, the student can compare the accuracy of the methods.

2. For a first order differential equation, use the plotter to sketch the vector field associated with it. The codes are not difficult to write and the student gains insight about the nature of solutions before obtaining explicit solutions or finding that no closed form solutions exist.

3. Additional assignments include using numerical information to guess solution techniques, to implement other solution techniques such as Milne's method, other Runge-Kutta methods, Adams methods etc, and to use the plotter to graph the isoclines for a given differential equation.

V. Projects on Infinite Series: The definition of convergence of an infinite series seems hard for a calculus student to understand. Thus several of the projects in this section should be done early in a unit on infinite series. The entire unit should be done rather late in a course of one variable calculus since some mathematical maturity is necessary.

Specific Assignments:

1. The most elementary but perhaps the most useful assignment in teaching convergence of a series of constants is to have the computer calculate and print several partial sums of a given infinite series. This will reinforce the idea that the value of a series is the limit of its associated sequence of partial sums. In addition students will be able observe that $\sum_{n=1}^{\infty}(a_n+b_n) = \sum_{n=1}^{\infty}a_n + \sum_{n=1}^{\infty}b_n$ provided both $\sum_{n=1}^{\infty}a_n$ and $\sum_{n=1}^{\infty}b_n$ converge, and similar results.

2. Students can use the computer to investigate the nature of functions which are given by an infinite series. Specifically, one might calculate a table of values for the exponential function $y = e^x$ by summing the series $\sum_{n=0}^{\infty} \frac{1}{n!} x^n$ Students will also learn about the radius of convergence and the interval of convergence of a power series by observing directly that something goes wrong with their computer programs if x is badly chosen in summing $\sum_{n=1}^{\infty}a_n x^n$.

3. One of the nicest and most informative uses of the plotter is to graph a function and its first few Taylor polynomial approximations. The student actually sees that the larger the degree of the approximating polynomial the closer the approximation of the function. A particularly nice example to plot is $y = \sin x$ on $[-3\pi,3\pi]$ and its successive Taylor polynomials. Figure 5 contains the graphs of sin x and its Taylor polynomials of degrees 5, 11 and 15.

4. The computer very nicely illustrates term by term integration and differentiation of series. A nice example here is to calculate $\int_0^1 e^{x^2} dx$ (since there is no closed form antiderivative for

$\int e^{x^2} dx$) by using the Taylor series for $y = e^x$, replacing x by x^2, and integrating term by term. One can use the computer to obtain the result with a high degree of accuracy.

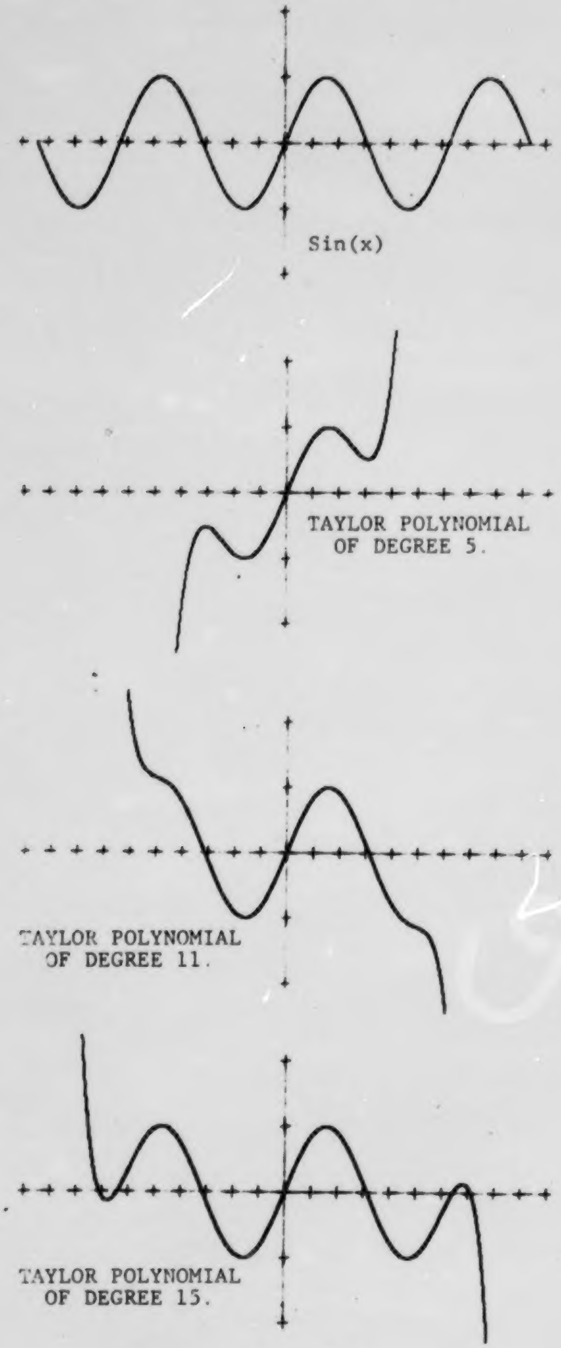


Figure 5

5. The value of π may be approximated by using the computer to calculate a large partial sum of $\sum_{n=1}^{\infty} \frac{1}{n^2}$, multiplying by 6 and taking the square root of the result. $\left(\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}\right)$.

Conclusions: The projects we have mentioned in this paper are some of those used successfully in our undergraduate calculus courses. The list is by no means exhaustive but rather is meant to convey the spirit of our computer oriented calculus. Every topic in calculus can be motivated or demonstrated by such projects even in schools with modest computing facilities such as ours. Perhaps the best measure of our success is in terms of student enrollment where even though our overall calculus population has increased slightly, enrollment in the computer oriented sections has more than tripled in the last 5 years and continue to increase rapidly. Such a progressive approach is invaluable to students entering a society in which some knowledge of computing is a requisite for every educated person.

CONSTRUCTING COMPUTER PROGRAMS FOR MATHEMATICS CAI USING A PEDAGOGICAL MODEL FOR TEACHING MATHEMATICS

Patrick W. Thompson
Thomas J. Cooney
University of Georgia

This paper will focus on the utility of a pedagogical model for teaching mathematics in constructing CAI lessons. The model is predicated on logical units of discourse, called *moves*, for teaching mathematical concepts and principles. Moves for teaching concepts include various types of characteristic moves and exemplification moves. Moves for teaching principles include introduction, interpretation, justification, and application moves. Each of the various moves is illustrated. Application of the model will be demonstrated for writing programs for teaching a mathematical concept and a mathematical principle, and for generating questions to assess students' understanding of mathematics.

Computer-assisted instruction (CAI) is becoming more of a vital force in instructional technology with each year. This trend is likely to continue and, indeed, accelerate given the advent of low cost mini-computers. To reach its full potential, however, authors of CAI programs must take care to insure that they produce programs based on sound pedagogy. To this end, this paper will focus on the explication of a pedagogical model for teaching mathematics and its application to developing CAI programs. It is our contention that the utilization of the proposed model will contribute to the advancement of CAI, and hence, to high quality mathematics instruction.

One of the major obstacles to the growth of CAI usage is the difficulty of creating pedagogically sound CAI computer programs. Our experience has been that this difficulty has its roots more in the requirements for conceptualizing the lesson to be presented via the computer than in the programming techniques requisite for its presentation. The modal approach to overcome this difficulty is one in which content free considerations are elaborated, possibly with examples, e.g., Yens, 1977; Burson, 1976. Therein one learns about such things as task specification, question/feedback construction, branching techniques, frame formatting, and frame flow. Each of these is an essential consideration one must take into account when constructing CAI programs. But as a collection they omit a vast area from which considerations critical to the pedagogical soundness of a program can be drawn, namely, the subject matter of the lesson. The balance of this paper will focus on the development of a model for teaching mathematics and its application to CAI.

Types of Mathematical Knowledge

B. O. Smith (1969) argues that each teacher should possess three types of knowledge: knowledge of subject matter, psychology, and pedagogy. To Smith, an important, but often neglected, aspect of pedagogical knowledge is the recognition of the various logical forms of the content being taught. To illustrate, Smith considered it essential that teachers differentiate among facts, laws, values, concepts, generalizations, and the like. When teachers can not make such differentiations, instruction suffers from vagueness and unnecessary ambiguities.

Cooney, Davis and Henderson (1975) identify three types of mathematical knowledge: concepts, generalizations, and prescriptions (or algorithms). Generalizations and prescriptions are collectively called principles. At the risk of oversimplification, a concept involves a nounlike term whose referent is a set of objects; a generalization is a truth-functional statement over some nonempty domain; and a prescription is a directive on how to perform some mathematical task. To illustrate, the following constitute mathematical concepts,

function
polynomial
triangle
prime number
parallel lines
equivalent equations

whereas the following constitute mathematical principles.

Generalizations	Prescriptions
1. For real numbers a and b $a \times b = b \times a$	1. To solve an equation of the form $ax + b = c$, $a \neq 0$, first subtract b from both members of the equation and then divide the result by a.

2. For real numbers x and y, $(x + y)^2 = x^2 + 2xy + y^2$.
3. If $a|b$ and $b|c$ then $a|c$.
4. In a right triangle with sides of length a, b and c where c is the length of the hypotenuse, $a^2 + b^2 = c^2$.
5. $\sin^2 x + \cos^2 x = 1$.

2. To determine the incenter of a triangle, construct the angle bisectors and find their point of intersection.

A Pedagogical Model for Teaching Mathematics

Once mathematical concepts and principles have been identified then one can consider the logical ways in which such content can be taught. These types of logical considerations are referred to as *moves for teaching principles* and *moves for teaching concepts*. This line of reasoning has led to a pedagogical model for teaching mathematics developed by Cooney, *et al.*, which consists of these moves. A brief elaboration of the moves is given below.

Moves for Teaching Concepts

Although logical operations, or moves, can be classified in various ways, the taxonomy presented by Cooney, Davis, and Henderson (1975) represents the most recent classification scheme. This classification scheme stems from the previous work of Smith (1967) and Henderson (1967). The moves were identified and explicated from an interaction of logical considerations and analyses of classroom discourse in which mathematical concepts were taught.

There are two general categories of concept moves: characterization moves and exemplification moves. Characterization moves provide information concerning the general nature of a concept's referent set. For example, characterization moves include necessary or sufficient conditions for an object to be an element of the concept's referent set and comparing and contrasting examples of the concept with other concepts. Exemplification moves involve the identification of examples and nonexamples, perhaps with justification why an object is or is not an example, and the use of counterexamples when a false statement has been made about the concept's referent set. Moves for teaching concepts using the concept of function are illustrated below.

I. Characterization moves

A. Necessary condition move. This move is designed to help the student learn information about a particular concept, that is, knowledge of the form "This is a ____________________, now what do we know about a ____________________?" The move states a necessary condition for the existence of a specific concept. The dialogue below represents a necessary condition move.

T: O.K. Now we know that this relation is a function. What does that tell us?
S: Well, we know each element in the domain must have a unique image.
T: All right. What else do we know?

B. Sufficient condition move. This move states a sufficient condition for the existence of a specific concept. That is, it provides knowledge of the form "If we know__________, then we know we have an example of a ______________. The dialogue below represents a sufficient condition move.

T: Can anyone tell me how we can determine if a given relation is a function?
S: If you graph the relation and any vertical line you draw will intersect the graph at most once, then it is a function.

C. Necessary and sufficient condition move. This move occurs when a concept or, more precisely, the term denoting the concept is defined. Necessary and sufficient conditions other than the defining ones can also be identified and discussed.

D. Comparing and contrasting move. The intent of this move is to identify similarities or differences between examples of the concept being studied and examples of other concepts perhaps previously studied by students. The following questions illustrate this move.

T: How does a linear function differ from a constant function? How are they similar?

E. Neither necessary nor sufficient condition move. Sometimes properties of a concept are identified which are important but are neither necessary nor sufficient for the existence of the concept. The dialogue below represents such a move.

T: Why are functions important for us to study?
S: We seem to run across functions in many different areas. We studied them when we had polynomials, when we studied trig, and other times too.
T: Yes. Functions can be a unifying theme in mathematics. For example, ---

II. Exemplification moves

A. Examples (with or without reasons) move. Examples of a concept can be given by either the student or the teacher. The example(s) may or may not be accompanied by reasons why the object(s) identified are examples. The following dialogue illustrates a student giving an example and the teacher then asking the student to justify his selection.

T: Who can give me an example of a function that is infinite but bounded.
S: That's easy. Just take a trig function.
T: Any trig function?
S: Well (after some thought), no, not all of them. But the sine function would be infinite and bounded.
T: Convince me.

B. Nonexamples (with or without reasons) move. Nonexample moves involve the identification of nonexamples which may be accompanied with reasons why the object is a nonexample. The question below illustrates a teacher posing a nonexample and then asking students to provide reasons why it is a nonexample.

T: We decided that a parabola of the form $f(x) = x^2$ defined a function. Why is the parabola defined by the equation $y = \pm\sqrt{x}$ not a function?

C. Counterexample move. This move is used to falsify a statement which represents a misconception. The dialogue below illustrates this move.

S: Functions are easy to identify. Every linear equation determines a function.
T: How about $x = 3$?

Moves for Teaching Principles

Moves for teaching principles are categorized into four groups: introduction, interpretation, justification, and either application or practice in using the principle.

As the term suggests, introduction moves deal with various ways a principle can be introduced to a student. This includes stating the principle or focusing students' attention on knowledge which might lead to a discovery of the principle. It also includes the identification of objectives for students to master in learning the principle. Providing motivation is also classified as an introduction move.

Interpretation moves are intended to help students understand what the principle means. They include investigating specific cases called instances or in the case of a prescription demonstrating an indicated procedure. Other means of interpreting a principle include pointing out the hypothesis and conclusion of a generalization or indicating contexts in which a prescription would apply. A student may also better understand a principle if some of the prerequisite knowledge for the principle is reviewed. Concept moves would be used to review concepts prerequisite to learning the principle. The remaining interpretation move is paraphrasing the principle. This move can be of several forms. It might entail students stating the principle in their own words or the teacher stating the principle in simpler language. This move also consists of stating or illustrating the principle with the aid of a diagram. This occurs frequently when the principle is geometric.

The third cluster of moves are justification moves. They are intended to help convince a student that a principle is true. One means of doing this is by investigating a number of instances. For generalizations, this entails replacing the variables in the domain of a generalization with constants and showing that the resultant statement (an instance) is true. For prescriptions, it involves demonstrating the prescription for a specific case and observing if the desired result is obtained. Another means of justifying a principle is to provide a deductive argument. Such an argument would be based on established definitions and principles. A third means of establishing a principle is searching for a counterexample. If a counterexample cannot be found, then the principle becomes more believeable. Of course, this method of justification, as is examining instances, is not a conclusive kind of argument. However, for the less mature student who is not capable of comprehending an axiomatic argument the examination of instances and searching for counterexamples may be the only means of justification other than relying on authority.

The last category of moves for teaching principles is application for generalizations or practice for prescriptions. Application involves using a generalization to solve a problem or exercise or to develop other mathematical principles. Practice refers to repeatedly following intended prescriptions. If practice is used with appropriate reinforcement and feedback, then a student will likely become skilled in following the prescription.

Moves for teaching principles are illustrated below using the following principle.

If $ax^2 + bx + c = 0$ and $a \neq 0$ then $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

I. Introduction moves

A. Focus move . An example of a focus move might be: Today we are going to learn how to solve a quadratic equation of the form $ax^2 + bx + c = 0$, $a \neq 0$.

B. Motivation move. This move usually consists of more than one statement. One means of motivating the learning of a principle is to pose a problem. For example, if students had learned to solve quadratic equations only by factoring, they might be presented with the problem of solving $x^2 + x - 1 = 0$. Subsequent discussions of how to solve this equation could serve as a motivating force to learn the above principle.

C. Objective move. Numerous objectives could be stated for learning the quadratic formula. One might be: The student should be able to solve any quadratic equation of the form $ax^2 + bx + c = 0$, $a \neq 0$, using the quadratic formula.

II. Interpretation moves

A. Instantiation move. The following statements are instances of the above principle.

i) $2x^2 - 3x + 1 = 0$ hence $x = \frac{+3 \pm \sqrt{9 - 4(2)(1)}}{4}$

ii) $x^2 - 5x + 3 = 0$ hence $x = \frac{+5 \pm \sqrt{25 - 4(1)(3)}}{2}$

B. Analysis move. The following questions are based on identifying the hypothesis and conclusion of the above principle. These questions could serve as discussion questions for students or as a means of ascertaining the student's knowledge of the principle.

i) If $2x^2 - 5x = 12$, what value does c have in the formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$?

ii) Why can you use $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ to solve quadratic equations and not linear equations?

C. Review move. The following questions could be posed to review prerequisite knowledge for learning the principle.

i) How do we determine the value of b in the equation $3x - 4 + 2x^2 = 0$?

ii) How do we find the square root of a number?

D. Paraphrase move. The following requests of students require a paraphrase of the given principle.

i) Elsie, tell me in your own words what the quadratic equation says.

ii) Now suppose $bx^2 + ax + c = 0$. What does the formula say x will equal in this case?

III. Justification moves

A. Instatiation move. The following dialogue represents the use of instances to justify the above principle.

Teacher: O.K., now when we solved $x^2 - x - 12 = 0$ by factoring what did we get?
Student: We got $x = 4$ and $x = -3$.
Teacher: Good and what did we get when we solved $2x^2 + 5x - 12 = 0$ by factoring?
Student: $x = -4$ and $x = 3/2$
Teacher: Fine. Now we are going to solve the same two equations by using the quadratic formula. Then we will see if we get the same results as we obtained when we solved the equation by factoring.

When the same solutions are obtained, this should help convince students that the principle is true.

B. Deductive argument move. The principle could be established by a deductive proof.

C. Searching for a counterexample move. This move could consist of having students create quadratic equations which they think might not be solveable by the quadratic formula. When solutions are obtained using the formula, then the principle will be more believable.

D. Authority move. This would simply request the students to believe the principle because the teacher or the textbook says it is true.

IV. Application moves

The following two problems could represent an application of the quadratic formula.

i) Solve the following equation for x. $x^2y^2 + 3x + 4yx + 6y + 12 = 0$

ii) Classify the roots of the equation $2x^2 + 4x + 13 = 0$ as rational, irrational or complex.

Applications of the Model

In this section the application of moves for designing CAI programs will be given. First, we will consider programs for teaching mathematical concepts.

Programs for Teaching Concepts

In general, the following questions can be posed for writing programs for teaching mathematical concepts.

1. What characteristics of the concept should be presented to the students? In particular, what necessary and/or sufficient conditions for the concept should be explicitly treated in the program? How and with what other concepts, can the concept being taught be compared and/or contrasted with other mathematical concepts.
2. What examples and nonexamples should be considered in designing the program? How should they be sequenced? Should examples and nonexamples be accompanied by reasons? Can counterexamples be used by asking students to consider false statements about the concept?

Once answers to the above questions have been ascertained, then a sequence of moves, that is, a teaching strategy can be designed for the program.

The strategy for teaching the lesson can then be embodied in a flow chart. In this context the branching capabilities of the program can be considered. If, for example, a student is having difficulty making sufficient condition moves, you may want the program to branch to a section which provides sufficient condition moves and distinguishes between sufficient and necessary conditions; it might also include a section that gives example with reason moves, the reasons being sufficient conditions.

Excerpts from a program which teaches the concept of equivalent fractions are given below to illustrate various concept moves. A copy of the flow chart for this program may be obtained from the authors.

Definition move

```
    The concept that we will work on today is 'equivalent
fractions'.  Two fractions a/b and c/d are said to be
equivalent if ad = bc .
```

Necessary condition move

```
    Notice here that we've shown that if two fractions are
equivalent, then when we reduce either to lowest terms, we
get the same fraction.
```

Sufficient condition move

```
Suppose c = k x a  and  d = k x b  for some non-zero integer
k .  Are a/b and c/d equivalent fractions?
```

Compare and contrast move

```
        How are equivalent fractions similar to equivalent equations?

           (a)  Equivalent fractions give different ways to represent
                the same number; equivalent equations give different
                ways to represent the same set of numbers (the solution
                set).

           (b)  It is two fractions that are equivalent; it is two
                equations that are equivalent.

           (c)  Equivalent fractions can be reduced to a common
                fraction; equivalent equations can be reduced to a
                common equation.

                  (1) a only    (2) b only    (3) c only
                  (4) a and b   (5) b and c   (6) a and c  ?
```

Nonexample with reason move

```
3/5 and 19/37 are not equivalent, since 3 x 37 <> 5 x 37 .
```

Example with reason move

```
27/36 and 18/24 are equivalent fractions because when we
reduce each to lowest terms we get 3/4.
```

Programs for Teaching Principles

The following questions can be considered when writing programs for teaching mathematical principles.

1. How can the principle be introduced? What objectives for the lesson should be stated? How can the principle be motivated?

2. How can the principle be clarified? What specific cases should be examined? How can the principle be analyzed? What prerequisite knowledge should be reviewed? How can the principle be paraphrased?

3. How can the principle be justified? Should specific cases be examined to see if the principle always works? Should a proof be presented? Should students be asked to search for counterexamples to the principle.

4. How can the principle be applied? What problems using the generalization should be presented? Can the principle be applied to generate other mathematical principles?

As is the case with concepts, your answers to the above questions can be used to design a teaching strategy to teach a generalization. This strategy can then be embodied in a flow chart for direct translation into a program.

Excerpts from a program developed from a strategy to teach a generalization are given below to illustrate various moves for teaching a generalization. The generalization is:

For natural numbers a and b, $(a + b)(a - b) = a^2 - b^2$.

For illustration, a list of the sequence of moves from which the flow chart and program were constructed follows the excerpts. A copy of the flow chart for the program may be obtained from the authors.

Assertion move (Give a statement of the generalization)

```
    The generalization (property) that we're going to
learn is:
                   (a + b)(a - b) = a↑2 - b↑2
```

Analysis move

```
Let's look at this generalization for a minute. Notice that
there are only two numbers involved -- namely a and b. Also,
notice that we really have two statements -- the one above
and the statement
               a↑2 - b↑2 = (a + b)(a - b)
   So now we know two things:  if we have a product in the
form (a + b)(a - b) we can write it as 'a↑2 - b↑2'.  Also,
if we have a number in the form a↑2 - b↑2 we can write it
in the form (a + b)(a - b).

Press RETURN when ready to continue ...
```

Paraphrase move

```
This is the generalization:
               (a + b)(a - b) = a↑2 - b↑2

Which of the following is another way to say the same thing?

 1 The product of the sum and difference of two numbers
equals the difference of the squares of the numbers.

 2 The product of the sum and difference of two numbers
equals the square of their difference.

 3 The sum of the product and difference of two numbers
equals the square of the difference of the numbers.

 4 The difference of two squares equals the product
of their sum and difference.

What is your choice? 4
```

Instantiation move

```
               (a + b)(a - b) = a↑2 - b↑2

                a = 23          b = 17


So we have:

23↑2 - 17↑2 = (23 + 17)(23 - 17) = 40 x 6 = 240

Press RETURN when ready to continue ...
```

Application move

```
               (a + b)(a - b) = a↑2 - b↑2


What is 24 x 16 ? 386
That's not right.  Try this same one again, PAT.


What is 24 x 16 ? 394
You missed this one.  Here's the way to work this out:

 24 x 16 = (20 + 4)(20 - 4) = 20↑2 - 4↑2 = 400 - 16 = 384
You'll do better next time.

Press RETURN when ready to continue ...
  Here's another one ...
```

Analysis move

```
Here is the product  ...  71 x 64
Can we use the generalization (1=Yes,2=No)? 1
That wasn't so good.
We would have to write the expressions using fractions (try it!)
Try another one ...

Here is the product  ...  40 x 32
Can we use the generalization (1=Yes,2=No)? 2
That wasn't correct.
 40 x 32 = (36 + 4)(36 - 4).
Try another one ...

Here is the product  ...  98 x 79
Can we use the generalization (1=Yes,2=No)? 2
Very Good !!
Try another one ...

Here is the product  ...  95 x 80
Can we use the generalization (1=Yes,2=No)? 2
You're correct !
Try another one ...

Here is the product  ...  96 x 87
Can we use the generalization (1=Yes,2=No)? 2
You're doing great !
```

Justification move

```
   The reason that the generalization:
               (a + b)(a - b) = a↑2 - b↑2

holds for natural numbers is because of the distributive
property of multiplication over addition.  Starting out with
(a+b)(a-b), we apply the distributive property, getting
        (a+b)(a-b) = (a+b)a - (a-b)b
(remember, the distributive property says that x(y-z) = xy - xz.
In this case x = (a+b).)
   Then we apply the distributive property again in the
right hand side above, getting
        (a+b)a - (a+b)b = axa + bxa - (axb + bxb)
                        = a↑2 + bxa - axb - b↑2
Since bxa - axb = 0, we have that (a+b)(a-b) = a↑2 - b↑2

Press RETURN to continue ...
```

Generalization - For natural numbers x and y, $(x + y)(x - y) = x^2 - y^2$

Strategy - Expository (Assertion appears early in strategy)

Focus move	Get the student's attention.
Motivation move	Explain that the generalization has some utility.
Assertion move	Give the statement of the generalization.
Analysis move	Point out salient features of the statement, such as that there are only two numbers involved.
Paraphrase move	Ask the student to express the generalization in words.
Instantiation move	Replace the letters by numbers. Extended commentary.
Instantiation move	Short commentary.
Instantiation move	Short commentary.
Application move	Have student apply the generalization.
Application move	Have student apply the generalization.
Instantiation move	Use larger numbers.
Instantiation move	Short commentary.
Application move	Have student apply the generalization.
Application move	Have student apply the generalization.
Analysis move	Supply product for which the generalization does not hold.
Analysis move	Have student determine whether or not the generalization may be used.

Analysis move	Have student determine whether or not the generalization may be used.
Justification move	Deductive argument.
Summary	Summarize the lesson and student's performance.

Using the Model to Evaluate Students' Understanding

Moves can also serve as a basis for designing questions to evaluate a student's understanding of a concept or generalization. To illustrate, consider the concept of rational number. Sample questions and the moves to which they correlate are given below.

1. Suppose x is a rational number. Can x have the representation 0.0n00n000n. . . for some digit n? (necessary condition move)

2. The graph of the function $f(x) = mx + 2$ passes through the points $(0,2)$ and $(\sqrt{3}, 2/\sqrt{3} + 2)$. Is m a rational number? (sufficient condition move).

3. Which of the following does not represent a rational number?

   a.) .135282828. . .
   b.) 17/34
   c.) .1234567891011. . .
   d.) .1234567891234. . .

   Why does your choice not represent a rational number?

   a.) It is not a fraction.
   b.) It is not in decimal form.
   c.) There is no pattern to the digits.
   d.) There is no block of digits that repeats. (Nonexample with reason move).

4. Which of the following represents a rational number?

   a.) $\sqrt{2} / \sqrt{8}$
   b.) .122333444455555. . .
   c.) $\sqrt{27}$
   d.) 4¼ (Example move)

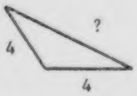
5. For each of the following statements tell whether it is (1) true for natural numbers only; (2) true for rational numbers only; (3) true for both

   a.) The solution set for the equation "$3x - 5 = 0$" is nonempty. (1, 2,or 3)?
   b.) Each number can be represented by the ratio of two integers. (1, 2 or 3)?
   c.) For every number x there is a number y, such that $xy = 1$. (1, 2, or 3)?
   d.) Between any two numbers x and y the collection of numbers between x and y is finite. (1, 2, or 3)? (compare and contrast move).

The questions below could be used to evaluate a student's understanding of the Pythagorean Theorem. The moves underlying the questions are given in parentheses.

1. Which of the following problems could be solved by a direct application of the Pythagorean Theorem?

   (a)

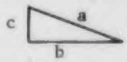


   (b) What is the longest possible diagonal in a rectangular prism 5 cm x 10 cm x 15 cm?

   (c) Is a triangle with sides of length 5 cm, 12 cm, and 13 cm a right triangle?

   1. (a) only 2. (b) only 3. (c) only 4. (a) and (c) 5. (b) and (c) (analysis move)

2. Which of the following is a statement of the Pythagorean Theorem based on the figure below?

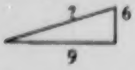


   (1) $a^2 + b^2 = c^2$ (2) $a^2 = c^2 - b^2$ (3) $b^2 + c^2 = a^2$ (4) $c^2 + a^2 = b^2$

   (paraphrase move)

3. Can you find a right triangle for which the Pythagorean Theorem does not hold? (Search for a counterexample move--a type of justification move)

4. What is the length of the indicated side? (instantiation or application move)



5. Which of the following shows how to obtain the distance in the x-y plane from (x_1,y_1) to (x_2,y_2) by using the Pythagorean Theorem?

(1) $d = (x_1 + x_2)/2 + (y_1 + y_2)/2$

(2) $d^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2$, so

$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$.

(3) $d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$, so

$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

(4) $d^2 = (x_1 - x_2)^2/2 + (y_1 - y_2)^2$, so

$d = \sqrt{(x_1 - x_2)^2/2 + (y_1 - y_2)^2/2}$.

(application or analysis move)

CONCLUSION

The utility of the model for teaching mathematical generalizations and concepts is most apparent when you consider the organization that it provides for conceptualizing lessons. After a clear picture of the logical structure of a lesson has been obtained one may utilize more programming specific techniques such as flow charting and frame formatting. We see the construction of a CAI program as a three step procedure: (1) conceptualizing the lesson; (2) embodying the structure of the lesson within an appropriate model for program structure; and (3) writing the program. This construction is a procedure of translation from the realms of pedagogy to the field of CAI. Models appropriate to step (2) are easily accessible to CAI authors. However, they deal with the structure of computer programs, and hence are content free . Any model dealing with step (1) must come from pedagogy, and in addition must account for the logical structure of the subject matter with which it deals. Content-free models are unable to account for the ideosyncrasies of specific subject matter areas, and hence offer little assistance in the crucial first step. The pedagogical model explicated herein gives a means to account for the structure of mathematical subject matter within a pedagogical framework, and hence it can assist CAI authors in producing quality mathematics CAI programs.

References

Burson, J. L. The author's guide to CAI (Vol. 1). Columbus, Ohio: The Ohio State University, Division of Computing Services for Medical Education and Research, College of Medicine, 1976.

Cooney, T. J., Davis, E. J., & Henderson, K. B. Dynamics of teaching secondary school mathematics. Boston: Houghton Mifflin, 1975.

Henderson, K. B. A model for teaching mathematical concepts. The Mathematics Teacher, 1967, 60, 573-577.

Smith, B. O., Cohen, S. B., & Pearl, A. Teachers for the real world. Washington, D.C.: The American Association of Colleges for Teacher Education, 1969.

Smith, B. O. & Meux, M. O. A study of strategies of teaching. Urbana: Bureau of Educational Research, College of Education, University of Illinois Press, 1967.

Yens, D. P. Manual for developing instructional materials for computer based instruction. Unpublished manuscript, Medical College of Georgia, 1977.

MODELING AND SIMULATION
AS A TEACHING TOOL

John J. Uhran, Jr.
William I. Davisson
University of Notre Dame
Notre Dame, Indiana 46556

Abstract

It is the conviction of the author's that a broad or universal approach to modeling and simulation is possible and practical. Further, this approach is discipline independent. In our experience both Engineering Students and Economics students (though with widely differing backgrounds) at the upper undergraduate level have adapted easily to the methodology and to the types of modeling and simulation assignments given. In addition, college teachers from many backgrounds have also found the method useful. The range of modeling and simulation assignments has been wide as illustrated by the two examples given in the paper.

Introduction

It is the authors conviction that the time for modeling and simulation as a more or less universal subject has come. Over the years, attempts to model and do simulation have been rather fragmented, independent, and very discipline oriented. Over the years, a strong and better understanding of system theory [1] and its relation to many other discipline areas [2] has come about. In addition, the evolution of the computer has had an enormous effect on every discipline; even those which have traditionally attempted to resist technological innovation such as in the arts.[3] A review of the literature indicates that in some areas such as engineering, modeling and simulation has become an important sub-discipline of study.[4] In most other disciplines, modeling has been attempted but the efforts have been fragmented.

This paper attempts to show that this is no longer necessary, that a universal approach to modeling and simulation can be accomplished and that the tools are presently at hand in order to let this happen. Indeed, a rethinking of people's approach to the problem can be accomplished.

The approach is a relatively easy one to take. We have accumulated enough evidence over the past two years to indicate that our challenge to you is in fact possible. Our efforts here will be to explain generally the approach used and to present a number of examples to show what is possible. Finally we will summarize our experiences in and outside of the classroom.

It is appropriate that we give recognition to the fact that the approach that we are using has come from J.W. Forrester of MIT using the ideas that he explained in his first book, Industrial Dynamics.[5]

The essential features of modeling and simulation methodology suggested here are three in number. First, it includes the idea, based on Control Theory, that any modeling and simulation problem can be formulated as a closed boundary system containing feedback which relates causes and effects. In fact, feedback is the method of establishing control in a system. Second, is the idea of a structured model. A structured model, as we regard it, is one wherein all of the variables are specified by the modeler, all of the relationships are identified (both process links and information links) and the data parameters or coefficients are indicated independently of the simulation itself. That is, the data coefficients are not generated by the simulation run as they are in other simulation methodologies but are instead constant.[1] Third, in this approach to modeling and simulation, emphasis is placed on the modeling of rate or decision variables that control the behavior of the simulation itself.

All in all, the characteristics of this particular methodology constitute its uniqueness in approaching modeling and simulation. Further those who disagree with this approach generally do so on the basis of the three points noted.

Modeling and Simulation

In order to understand the discussion below, it is necessary that we all understand the nature of modeling and simulation. In addition, it is necessary to realize that modeling and simulation are complementary to each other, that one without the other is like an automobile without engine or wheels. In our minds modeling and simulation are so interrelated that one must cooperate with the other.

---

[1] In Economics and statistics, a model is called structural because the coefficients have yet to be determined. In this method the term structural is used to emphasize that the coefficients are introduced as constants and are not determined by the simulation itself. In addition, in this method, the idea of a structural or structured model in which the modeler determines the relationships among the variables is essential.

Modeling. A model is a representation of the real world. Whether a model be physical in nature such as a model airplane or mathematical in nature such as a set of equations or relationships, they all have the same function; that is, attempt to replicate what in fact exists in the real world. There is indeed a good reason for being able to model and model well. Not only do models allow us to have a better understanding of the world we live in, but they are also far more efficient and less costly in many ways. It is easier to build a model plane and test its performance in a wind tunnel than to actually build a physical model to obtain the same information. It is easier to create a model of queueing for making a court system more efficient than to actually have to try certain policies in a real court system over a long period of time. Indeed there are times when modeling is almost necessary since the real system cannot actually be experienced ahead of time, such as modeling the landing of a vehicle on mars or the moon before the actual trip took place. As with anything involving approximations, modeling indeed has its limitations. While modeling has been relatively well used and tried in the engineering disciplines, and even in science, it has been less commonly used but nevertheless attempted in business, social sciences and the humanities. It is certainly not universally accepted or recognized as a valid tool in many disciplines, but it is a way to open up and create new visions for the discussion of many interesting and important problems that cross disciplinary lines and give man a new vision of himself and the world he lives in.

Simulation. Simulation, the counterpart of modeling, is the exercise of a model. While modeling is in and of itself important and a useful learning experience, it is simulation which essentially completes the process which has been started. It is here that we learn in detail, how our model works, how it can be changed and modified and in effect what are its limitations. The modern computer and microcomputer have certainly helped to generate more interest in simulation and has helped to advance the state of the art in modeling and simulation.

Continuous	Discrete
DYNAMO (1959)	SIMSCRIPT (1962)
CSMP	GPSS (1961)
ANSIR	SIMULA (1965)
DSL/90	GASP
MIDAS	GERTS
ECAP	
NDTRAN	

Table 1.

It should be remembered that there are various methods of simulation such as model plane, discussion, or, in fact, mathematical solutions using the digital computer. In order to aid the modeler a number of computer simulation languages have been developed over the years. Table 1 indicates the most common simulation languages which are presently available and the type of simulation that is appropriate-continuous and discrete-and the year when the language became available.

Method of Approach

The question that the authors sought to ask each other and their colleagues was, "Is there a universal way of attacking modeling and simulation problems?" In other words, is there a universal type of simulation language that would be disciple independent; that would allow models and the testing of models to be accomplished but yet would leave open all the previous choices developed by those who are discipline specific? Is there a way in which the emphasis is taken off programming and put in the proper perspective of modeling? We think that we have found a way to accomplish these goals.

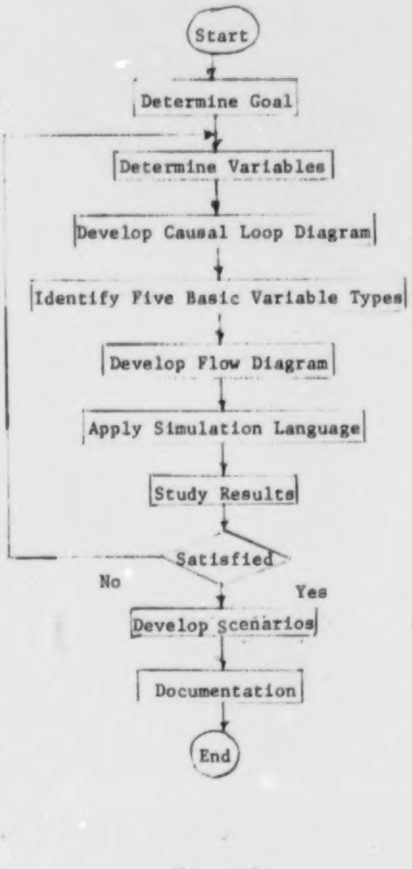


Figure 1.

Realizing that modeling and simulation should go hand in hand we have taken some basic ideas from industrial dynamics, and evolved it into a way of developing a problem from the beginning to the end. This methodology goes through the entire process of modeling and simulation and allows for a complete answer. What we have attempted to do is evolve a method which makes the concept of modeling more universal and provides at the same time an easy to use simulation language that makes the solution of the models created understandable. The methodology or approach here is one that can be universally taught in most courses and is indicated in Figure 1. Each step in the process evolves from the previous steps and is logically consistant. The way in which each step is applied to a specific course is the difference which we believe has created the universality. We believe that the method can be as easily applied in social systems, especially to those problems where no mathematical expressions are known to exist. However, those who have sophisticated mathematics background such as in engineering and science have no less difficulty or advantage in using this particular approach.

Methodology by Example

We believe that the only way to learn how to model and simulate those models is to actually do the process. It is not only essential that reasonably good examples, simple to understand, are initially presented at each stage of development but that good problems should evolve so that the sophistication of the method can be exploited by the individual at the end of a reasonably short period of time. In addition, it must be emphasized that any problems you should be bold enough to try should encompass either engineering, scientific or social science disciplines.

The modeling and simulation of any problem using this approach has three basic steps in common; (1) Developmental Stage which is graphically portrayed by a causal loop diagram. This is simply a graphical way of portraying the interaction between the selected variables of the problem. It tends to indicate which variable effects any other variable and whether the effect is negative or positive. As will be seen, this usually creates loops which introduce very effectively a most important concept called feedback; (2) The development of a flow diagram. Here the variables are now characterized into various categories, such as rates, levels, auxilliaries, constants and so on. This particular step introduces a more specific structure into the problem as is common with simulation methods frequently used in engineering; (3) The final step is the actual writing and execution of a computer program to perform a simulation of the model developed.

We believe that these three steps are common to almost any modeling/simulation procedure and involve a common sense approach to the process. If a particular method of simulation at the end of this process is required, then both the flow diagram and the information that was brought to bear upon it can be effectively translated to any computer simulation language that we are presently aware of. (Refer to Figure 1 for guidance).

In the remaining part of the section, we would like to indicate how these steps are followed through specific and quite diverse problems and give some insight to the reader concerning the methodology involved. While each of the problems can be made more complicated, for purposes of illustration and simplicity, we will attempt to describe only the bare essentials, so that the technique and the classroom value can be clearly demonstrated. We urge the reader however to keep in mind that there is a distinct difference between the two problems. The first one is a social science problem which is basically unstructured and in general has no natural mathematical approach to a solution. It is also typical of the type of problem which evolves in a social science environment and which in fact can have many answers. On the other hand, the second example is very mathematically structured which an engineering student should be quite capable of solving analytically without going through the modeling process. Nevertheless the student might at some time want to see a complete dynamic solution or simulation and thus the modeling process for him is equally valid and equally useful.

Example 1: Simple Population Model.

Problem statement: In a certain community, the population is determined by the birth rate and the death rate. The birth rate is a constant 5 persons born/year/thousand people in the community. The initial census indicated that one thousand persons inhabited the area. Determine the population after 500 years if the death rate is a constant 4.5, 5, or 5.5 persons/year/thousand people.

While there are many factors which affect population dynamics, the three most critical variables are the actual population itself, the rate at which the population decreases due to deaths, and the rate at which the population increases due to births. Factors such as attitudes, disease, famine, etc. are only implied. We can thus draw the following causal-loop diagram.

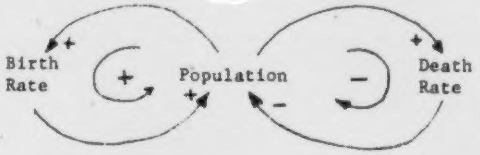


Figure 2.

The causal-loop diagram reflects the fact that the population level will affect the actual number of births and deaths per year, which in turn affects the actual population level. The signs have been added to the diagram to indicate direction of influence. In the case of the birth rate as the population increases, the number of births increases, which in turn has a positive effect causing the population to increase more. On the other hand, the level of population will affect or increase the number of deaths but this in turn, will have a decreasing negative effect on the population level. Thus, one of the signs here is positive and one of them is negative. Common sense indicates that the population is a level which will change whenever births or deaths occur. These variables are actually rates, that is, they are the rate of changes of the population, either increasing or decreasing it. The actual birth rate or death rate constant for any given population, whether it be the population in a city, country or the world, can be found from statistical data, usually in so many per thousand people.

Two very important ideas should be discussed at this point. First let us assume that there are no deaths. From the diagram and the concept of population, it is clear that since the population is at a certain level it will generate a certain number of births in one year. Keeping the birth rate constant will cause more births to occur each succeeding year, that is the rate of increase is constant but the incremental change is increasing. Thus we have the basic principle associated with compound interest which is exponential growth. The same idea would hold true if birth rates were set to zero and the death rate were the only operative variable. But in this case, the population would decrease each succeeding year in smaller and smaller increments and would exponentially drop to zero. Note that when neither the birth rate or the death rate is zero, one will counter the other to a certain extent, but the population will still increase or decrease exponentially. If the birth rate and the death rate were exactly the same, then no change in population could take place and we would have zero population growth. In these ways alternative scenarios maybe examined. A flow diagram for this example is shown in Figure 3.

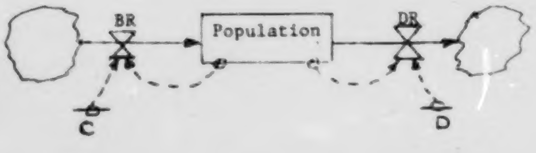


Figure 3.

In this figure, we have identified the population as a level, that is, a variable which only changes periodically. It is assumed that the flow of population comes from an infinite source and goes to an infinite sink, and that the rate at which the flow from the source into the sink takes place is controlled by the death rate (DR) and the birth rate (BR). The dotted lines on the diagram indicate not actual flow of people or things but simply information which affects the birth rate which controls the actual flow of people through the system.

A computer program that would execute this problem in NDTRAN[2] is given in Figure 4 below.

```
◆ ONE LEVEL POPULATION MODEL
◆NOSTATS
◆NARROW
◆EULER
L P.K=INTEGRAL(BR.JK-DR.JK)
R BR.KL=C◆P.K
R DR.KL=D◆P.K
C C=.005
C D=.0045
N P=1000
SPEC DT=1,START=0,STOP=500,PRTPER=20
PRINT P
RERUN
C D=.005
RERUN
C D=.0055
```

Figure 4

Notice that additional information has been supplied in the program for proper execution. These include specification for printing and plotting, constants (C) and initial values for levels (N). DT represents the inter calculation period.

```
              DR=5.5    DR=5.0       DR=4.5
     TIME       P         P             P

     0.00    1000.0    1000.0       1000.0
    20.00     990.0    1000.0       1010.0
    40.00     980.2    1000.0       1020.2
    60.00     970.4    1000.0       1030.4
    80.00     960.8    1000.0       1040.8
   100.00     951.2    1000.0       1051.3
   120.00     941.8    1000.0       1061.8
   140.00     932.4    1000.0       1072.5
   160.00     923.1    1000.0       1083.3
   180.00     913.9    1000.0       1094.2
   200.00     904.8    1000.0       1105.1
   220.00     895.8    1000.0       1116.2
   240.00     886.9    1000.0       1127.5
   260.00     878.1    1000.0       1138.8
   280.00     869.3    1000.0       1150.2
   300.00     860.7    1000.0       1161.8
   320.00     852.1    1000.0       1173.5
   340.00     843.6    1000.0       1185.3
   360.00     835.2    1000.0       1197.2
   380.00     826.9    1000.0       1209.2
   400.00     818.7    1000.0       1221.3
   420.00     810.5    1000.0       1233.6
   440.00     802.5    1000.0       1246.0
   460.00     794.5    1000.0       1258.5
   480.00     786.6    1000.0       1271.2
   500.00     778.8    1000.0       1283.9
```

Figure 5

---

2. This program has been written for the NDTRAN interpreter. It is similar but not the same as for DYNAMO.

A table of the population levels for the 500-year period beginning now is given in Figure 5 for the three death rates indicated. Note the exponential behavior except in the case where the two rates are equal.

The evolution of this problem is as follows; A problem definition is given; the key variables are defined, causal relationships are established; and parameter assignments are made. This is typical of the approach that must be taken to solve all social science problems. If mathematical equations are available to establish relationships, all to the well and good. But this is not necessary, since the method demonstrates that the causal relationships are sufficient to set up the dynamic nature of the problem.

Example 2: Tank Problem

Problem Statement: As indicated in Figure 6 a cylindrical tank filled with a liquid has a small orifice at the bottom of the tank where the liquid will flow out at a certain rate. There is the possibility of allowing for the flow of liquid into the tank as well. Given the area of the cylinder (A), it is required to compute the volume of water in the tank as a function of time.

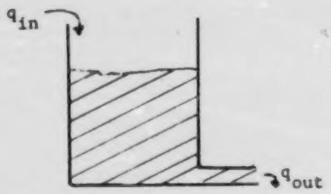


Figure 6

This is a simple problem which a student in Physics or Engineering might be faced with in a mechanics class. It is easy to identify that there is an input variable to the system which we identify as q_{in} and an output variable q_{out} which represents the flow of water in and out respectively in cubic feet per unit time. With the problem defined as above, the goal of the problem has been specified to determine the volume as a function of time. A causal loop diagram may be drawn as shown in Figure 7 below.

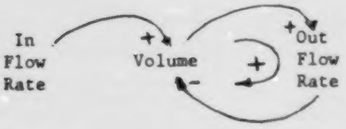


Figure 7

The causal loop diagram identifies the volume variable as the center of the problem. Clearly volume is affected very directly by the input flow rate, q_{in}, and the outflow rate, q_{out}. However, there is one additional link on the causal loop diagram so far not directly discussed. An astute student will realize that the volume does in fact affect the output flow. That is as the volume decreases the flow rate will tend to decrease and vice versa. In this simple example we will assume that the flow rate out is directly proportional to the volume and indirectly proportional to the area and some constant associated with the size and type of orifice. This is a simplifying assumption and a student could in fact find the exact relationship if it was necessary. As in the previous problem, we are able to assign signs to each cause and effect and to see that clearly we have one feedback loop which is negative in sign simply because it will cause the system to eventually go to a stable point. On the other hand the inflow rate is an exogeneous variable not affected by the system itself.

An exact analytical solution can be obtained from this problem without going any further. In fact, the differential equation describing the systems performance is

$$\frac{dv}{dt} = q_{in} - q_{out} \quad ; q_{out} = V/RA$$

At this point, an engineering student would be most likely to say that the solution is

$$V(t) = V_o e^{-t/RA} \quad t > 0 \ , q_{in} = 0$$

recognizing that we have a first order linear differential equation. However, those who are not familiar with differential equations would not even attempt to write the analytical solution but would go directly from the causal loop dia- to the flow diagram which appears in Figure 8.

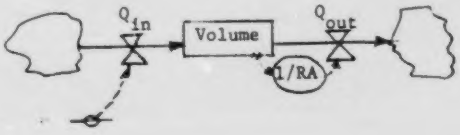


Figure 8

We now have a diagram which can be easily implemented in a number of simulation languages. If the causal loop diagram is done properly, then the flow diagram follows almost immediately. For those who are looking for a non-analytical solution but know the differential equation as above, the flow diagram provides an easy way for setting up the equations which will eventually be solved in a particular simulation approach. The flow diagram clearly indicates that the volume is controlled by a rate variable which controls the flow of liquid into the system. This is exogeneous, since no loops appear at that side of the figure. On the left side, the flow of water out of the volume variable is indicated by a rate variable which is, in fact, a function of the volume itself. Conserved flow of liquid in and out of the volume is indicated by the diagram. Figure 9 contains

```
* TANK PROBLEM
*NOSTATS
*NARROW
*EULER
L V.K=INTEGRAL(RVI.JK-RVO.JK)
R RVO.KL=(+1/(R*A))*V.K
R RVI.KL=RV.K
A RV.K=RVC
C RVC=0
A H.K=V.K/A
N V=10
C R=.5
C A=2
SPEC DT=.1,START=0,STOP=8,PLTPER=.4,PRTPER=.5
PRINT V,H,RVI,RVO
PLOT V=V
RERUN
C RVC=2
```

Figure 9

a listing of the program which can be implemented in NDTRAN directly from the flow diagram. A table of printed values for the solution and a plot of the volume versus time is included in Figures 10 and 11.

```
        TANK PROBLEM

   TIME          V          H              RVI        RVO

   0.0000     10.000     5.0000              0     10.000
   0.5000      5.905     2.9525              0      5.905
   1.0000      3.487     1.7434              0      3.487
   1.5000      2.059     1.0295              0      2.059
   2.0000      1.216     0.6079              0      1.216
   2.5000      0.718     0.3589              0      0.718
   3.0000      0.424     0.2120              0      0.424
   3.5000      0.250     0.1252              0      0.250
   4.0000      0.148     0.0739              0      0.148
   4.5000      0.087     0.0436              0      0.087
   5.0000      0.052     0.0258              0      0.052
   5.5000      0.030     0.0152              0      0.030
   6.0000      0.018     0.0090              0      0.018
   6.5000      0.011     0.0053              0      0.011
   7.0000      0.006     0.0031              0      0.006
   7.5000      0.004     0.0018              0      0.004
   8.0000      0.002     0.0011              0      0.002
```

Figure 10

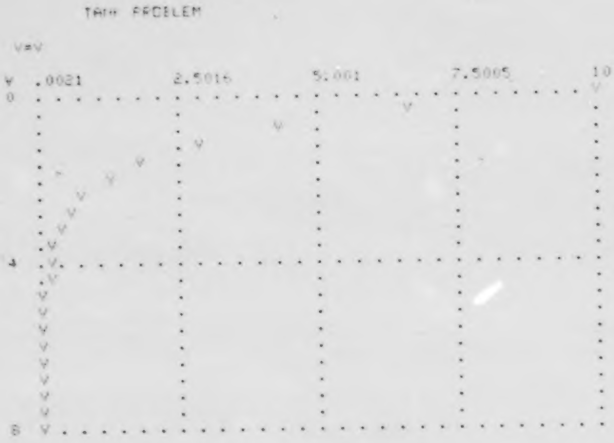


Figure 11

Scenarios for the problem can be done in a number of ways such as by changing q_{in} and making it either a constant, zero or some arbitrary function determined external to the problem. Since, in actuality, the outflow rate is really a non-linear function of the volume, not as assumed, the user could find the correct non-linear function in this case to find the answer. Once the problem has been set up and tried as was done here, many other different things can be attempted within the same environment. Once the simplicity of the method is understood as in this brief analysis, it comes very easy to extend the method to more difficult problems. The most difficult thing that the authors have encountered in attempting to teach modeling and simulation to students is how to make a problem simple rather than complex so answers can be initially found for analysis.

Computer Implementation

The above two problems have presented solutions which were done using a simulation language called NDTRAN.* This language has been developed at the University of Notre Dame in order to provide the simplest possible simulation approach to the general modeling methodology presented above. It is a computer language which is described in great detail in other places [6] but which was developed specifically for more universal use by being written in ANSI FORTRAN IV and being capable of running on any machine with a basic basic 16 K Core capacity and direct access disc. It has been successfully used in over 75 installations in small and large colleges and the computers which so far have been successfully able to use the NDTRAN package are listed in Table 2.

Table 2

NDTRAN - Proved Operations

- IBM 1130
- GA 18/30
- IBM 360 & 370 (DOS & OS compilers)
- DEC 10
- DEC 20
- PDP 11 (PDP 11/40 & 11/70 RSTSE)
- Harris System 210
- Xerox 560
- Xerox Sigma 6
- Xerox Sigma 9
- Univac 70/7
- Univac 1110
- GA SP 16/40
- HP 3000

With its ability to fit on a wide variety of machines, excellent diagnostic capability and a number of other features it is hoped that the smallest college, university of high school will now be able to avail itself of this particular technique.

---

* Developed from principles stated by J.W. Forrester in his early works.

Experience with this Approach.

The authors have had the unique opportunity to try this methodology out both at the University of Notre Dame in a multidisciplinary course and in a seminar program offered by the AAAS. At Notre Dame the authors have been able to teach this course four times to an interdisciplinary group of engineers and economics students generally at the senior level. From this course we have been able to develop a series of notes which we find are very helpful in the teaching of the material discussed. We believe that the students have learned a great deal. Many of the students in the class have been exposed to one or all of the types of problems that the course involves. The engineering students seem to adapt extremely easily to the general modeling approach, one that is frequently unfamiliar to them from their prior training. In turn, the economics students are able to adapt relatively easily to all of the various case studies presented, however, those problems which are very mathematically oriented do present some difficulties in terms of trying to understand the concepts. We feel that the course has been successful and with new materials now available we will continue to achieve its goal.

The authors have also had a unique opportunity to be able to discuss this particular method with over 200 college teachers. Approximately 85 of these have chosen to implement this approach in an existing course or develop a new course incorporating modeling and simulation in their courses. We feel as though we have been successful in achieving our goal in this regard.

Conclusion

The authors, having worked with a number of simulation languages in the past, and understanding the difficulty of teaching modeling to others, have become convinced that the above approach is one of the best available at this time. Our feelings have long been born out by the acceptance with which our work has achieved in the past several years and the ease with which we have been able to teach the method to both other teachers and students.

References

[1] Miles, R.F. ed, Systems Concepts: Lectures on Contemporary Approaches to Systems, Wiley Interscience, Somerset, N.J. 1973.

[2] Wiitaven, W. "Modeling Biological Systems by Means of Florugraphs," Simulation, v.26, June, 1976, pp. 185-192.

[3] Chitke, S.D. and Levis, A.A., "A Dynamic Simulation Model of Urban Litter Generation," IEEE Transactions on Systems, Man and Cybernetics, v. SMC-6, May, 1976, pp. 374-377.

[4] Chen, K. et al ed., "Social Systems Engineer-
in ing: An Introduction", IEEE Proceedings, v. 63, No.3, March 1975.

[5] Forrester, J.W., Industrial Dynamics, MIT., Cambridge, Mass. 1961.

[6] Davisson, W.I. Uhran, J.J., Primer and Problems for Systems Dynamics Interpreter (NDTRAN) Notre Dame, Indiana 1977.

COMPUTER ASSISTED INSTRUCTION AND HISTORY

Jane H. Wilde
Voorhees College

The usual reaction of historians to the computer is universally summed up in the word "quantification". Without underestimating the value of the computer in handling quantities of data not otherwise amenable to easy interpretation, to use the computer only for such purposes is to under-utilize a valuable and powerful adjunct to instruction. Programs, especially simulation, involve the student as a participant in the historical process. In developing the concept of the under-utilization of the computer as a teaching tool, the advantages and disadvantages of CAI are stressed. Some of the common objections to such programs are discussed. The papers surveys some of the literature relating to CAI in history, and comments briefly on some of the programs that have been developed.

Since the advent of the computer era, historians appear demonstrably to have become more interested in the computer as a powerful tool for the handling and interpretation of quantities of numerical data. This is not surprising. The early work in the field of quantification by historians such as Merle Curti and his students at the University of Wisconsin in the mid-1950's led to the collection of statistical data that could be more readily stored and utilized through the computer. From small beginnings such as this to the mass attention focused on quantification with the publication of Fogel and Engerman's controversial Time On the Cross, historians have come to think of the computer in relationship to history as a synonymous with quantification. Indeed, the publication of Time On the Cross resulted in much heated but fruitful examination of the strengths and weaknesses of the quantitative method as a historical technique.

The purpose of this paper, however, is not to debunk quantification or to deny its validity as a historical tool or as one type or a useful computer program for students. Indeed, my own research into the question of safety at sea as related to the experiences of the British merchant could have profited from such analysis. But the purpose here is to discuss the uses historians have made of computer assisted instruction. It is not meant to be exhaustive and ignores some rather obvious elementary cautions that could be given to historians as neophyte programmers.[1] A general survey of educational literature shows that this use of CAI has been modest indeed. Estimates indicate that perhaps only 50 historians are presently active in developing and using computer programs.

The advantages and disadvantages of CAI are much the same for historians as they are for most humanistic and social science disciplines. It is apparent that historians who have developed CAI programs or who have worked with quanitfication have been forced to rethink in greater depth their own function in the learning process. Until recently, with the onset of declining enrolments in history in particular and in college in general, historians have not notably focused much attention on learning theory. This, of course, reflects the whole pattern of graduate education as one which is oriented primarily to research and publication rather than teaching. This points to what has been identified as probably the major obstacle to greater use of CAI in higher education - faculty resistance.[2]

This factor seems even stronger among historians than among faculty in other disciplines. Perhaps the comments of Profs. Burns and Pycior at Notre Dame concerning the reactions of the history departments to a computer tutorial program in Western civilization is illustrative. They report a minimal impact of the project on historians not involved in the project, stating that attitudes ranged "from interest to open hostility. Most were indifferent. Preoccupied with their own teaching and research, even the most sympathetic were unwilling to invest the time and effort required to design and write tutorials".[3]

Perhaps three of the more common reactions of historians to CAI heard at sessions designed to introduce historians to CAI - the NSF funded program on Educational Computing in Minority Institutions, for example, are "it's a gimmick", "it takes too much time", and "it deals with fragments of knowledge, with bits and pieces". The "it's a gimmick" reaction arises mainly as an objection to the fact that the computer does appear to produce greater student interest in history . Perhaps it is felt that this interest is only transitory. It probably also reflects a degree of insecurity and a fear that the computer will diminish the role of the history teacher, creating further problems for a discipline already faced with shrinking enrollments and with the feeling that students often consider history to be irrelevant. However, the computer does involve the student in the historical processevery time the student goes to the computer terminal. It forces him to interact with the program. The computer provides different and more efficient ways to deliver historical data, involves the student in the analysis of data that would otherwise be too cumbersome to handle, can develop role playing abilities, and can simulate parts of the historical past. If this is so, then why object? It would appear that "it's a gimmick" loses its validity as anargument when greater student interest is created and students become involved with the material. After all, that is precisely what the historian wants in the first place.

The second argument, "it takes too much time" has far more weight than the gimmick argument,as any historian who has worked seriously with the development of a computer program is aware. It does involve a deatiled analysis of the sequence of the learning process and of the subject matter to be programmed. In addition, it involves learning a computer language. It also involves a continuing monitoring and revision of the program, once it is developed, to make it a more effective instrument.The time factor could be reduced somewhat if a partnership could develop between historians and computer programmers, which would obviate some of the need for historians to become experts in tha language of their computer. It is still important that the historian should know the basics of simple programming and the essentials of a computer language, for it will give hima greater insight into what the computer can do. However, the major contribution of the historian should grow out of his knowledge of history. He should also be awars of the common misconceptions of students, of what answers they are likely to give to questions. He should have analysed what entry level knowledge and what abilities the student needs to deal with a given problem. He should be able to design a model for the part of the past with which he is working. Perhaps this is saying nothing more than that the historian should be able to develop the flow chart for whatever he wishes to program. Once this is accomplished, he can then work with the computer programmer who can translate the program into the computer language.

Whether this sort of partnership is more of a present day description of how things might work, given the state of the profession, or whether it is a long range view is hard to answer. As we look at contemporary society, it would appear to be one of the tasks of the historian to put the development, use, and current state of computer usage into perspective. But it also appears that computers are here and will continue to play an increasing role in the educational process. Thus more persons in the future who become historians will already have the requisite computer languages to do their own programming.

The third objection, "it deals with bits and pieces of information" also has a foundation in fact. It is true that some of the computer uses that have been developed in history are test banks containing multiple choice questions. The use of test banks to generate different sets of test questions on a random basis, the greater ease of item analysis, and more rapid scoring are not to be overlooked asa powerful means of developing better tests. Other CAI programs have been drill and practice. These types of material have tended to create a strong impression of dealing with fragments. The average multiple choice - commonly referred to as "multiple guess" - questions generally does not require a high level of reasoning to arrive at the correct answer. This does not need to be so. Multiple choice problems - questions,if you will - can be worked out in such a way that the student must bring some additional knowedge and his reasoning powers to bear in order to arrive at the correct answer. It could not be answered if the student had only memorized facts. Of course, it takes time to develop such questions. This raises again the question of how the historian should spend his time. But the fact remains that programs do exist that appear to the historian to be rather simple-minded. Certainly that is not the fault of the computer nor an inherent defect of CAI, but one that can be remedied by the developers of programs. However, it might also be noted that there are certain basic bits of information that historians (and society) seem to expect educated persons to know. So some testing quite legitimately will deal with fragments of information, although the expectation is that these fragments will not be the end product of education.

It might be further noted that, of necessity, drill and practice does play a significant role in instruction. Much of an instructor's time is often taken up with the review of facts basic to the discipline, often utilizing class time that could be better spent with concepts and long range trends in history. In the area of drill and practice, the computer can become a patient and even tempered assistant who never wearies dealing with the same material time after time. It might be a more effective time use for the instructor to develop drill and practice tutorial, for in the long run it would conserve time and eliminate some of the routine dullness from his classroom.

In addition to its advantage as a personal and patient drillmaster, the computer offers other advantages to the instructor and to the learner. One of the points most frequently cited, especially by behaviorists, is the value of immediate feedback which helps quickly to erase misconceptions. Also, the computer does not publicly embarrass the weaker student who may be reluctant to reveal his lack of knowledge to his peers when he does not know the answer. The dialogue is between the student and the computer, which may, of course, keep records of his progress for the instructor, but no one else need know. Incidentally, the information thus gained about the student and his level of knowledge can further assist the instructor in meeting the needs of individual students. The computer also involves the student in an interactive dialog with the program, or in the last analysis, with the program, author, and if the student sharpens his knowledge to try to beat the computer, neither the outcome nor the means should be faulted. A computer can be skillfully programmed to have responses to typical student comments and wisecracks. It can even provide a variety of responses for a given comment. The creative historian will devise a program that is personalized to his own student population. He will continually monitor the program and study student responses stored in the computer for which the program has no appropriate answers in order to provide the needed responses.

One of the bigger tasks facing the historian using CAI is the question of how to interface computer programs with the instructional process so that the maximum benefit is obtained for the learner. This problem does not always surface in discussions of the use of CAI in history. It is a possible pitfall that deserves attention since it can have a negative effect on faculty attitudes toward CAI. The developers of a history tutorial at Notre Dame found this was an important problem they had ignored or underestimated - the failure to think through exactly how the CAI programs would relate to the rest of the course.[4] At least one instructor found the course frustrating although students reported high levels of satisfaction. The major problems lay in the failure of faculty to perceive the role of these tutorial programs. If they were viewed only as supplementary to the course, problems arose, since faculty often failed to realize the information delivery capacity of the tutorial programs. The failure to rethink thoroughly the goals and methodology of the course with a view to making the tutorials an integral part of the course was at the heart of the problem. Consequently the instructor felt that he has lost control of the course, that his role was not clear and that his importance was diminished. Thus the significance of this problem cannot be ignored. Stephen E. Lower of the University of British Columbia, although writing about CAI and general chemistry, has a lucid discussion of this very point. Indeed, he feels that interfacing CAI materials with the course is as important to the success of CAI as the CAI materials themselves.[5]

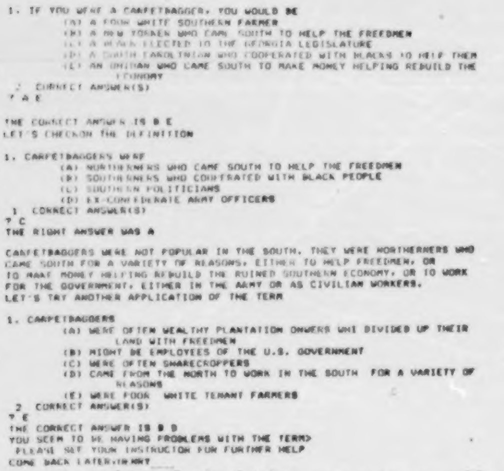
An additional question that comes to mind immediately if one is at all concerned with cognitive styles is whether CAI is really equally effective for all students and whether it is in line with their learning styles. Not enough is known about the learning process in general and cognitive styles in particular to give any definitive answer at this point.[6] However, it is a problem that deserves further study as one considers the use of CAI in the learning process. Enough is known to suggest that students do have different learning styles and teachers have different teaching styles. Thus educators should be seeking a more effective match between the two. CAI may, in fact, be helpful in forcing the rethinking of the structure of the discipline or subject matter and to identify different paths toward understanding it. It has been observed that the It has been observed that the structures associated with many disciplines are partly a result of the ways in which they usually have been taught. To quote Prof. Lower: "Because CAI can introduce so many new dimensions to learning, it is important that we take a careful look at our own subjects from these different viewpoints. Ultimately own would hope that it will become possible to emphasize different kinds of structural development that will correspond more closely to the needs of individual students. As one moves to more sophisticated terminal devices that are able to produce graphics and film displays, animated motion and audio responses, the possible ways of structuring the subject increase enormously, and present the possibility of developing alternative course structures attend to different cognitive styles".[7]

It should also be noted in passing that the development of programs using graphics should not be overlooked in the production of CAI history programs. Thus far, historians have made little use of graphics to enliven a computer program.

Beyond drill and practice and tutorial applications, the computer offers much in the way of role playing and simulation. In many respects, the historian in analysing and interpreting his material must be able to place himself in the context of the past he is studying. However, simulation is not always widely used in history classes. In simulation, the computer can be used to provide for the historian something analagous to the laboratory of the natural and physical sciences. The computer offers more flexilibity and is less cumbersome than the historical games that are sometimes used in history classrooms, A computer game can speed up the handling of mathematical and statistical data and the game can be more comprehensive.[8] The experience of the Huntington Project suggests that there is a demand and desire of the sorts of simulation games that have been developed.[9] However, most of the social studies programs developed under this project focus on concepts and processes in economics. These programs were designed for the high school level primarily. Nonetheless they can profitably be integrated into a variety of college social science courses.

The values and usefulness of simulations occurred to me particularly when I was teaching the second semester of American history survey. Dealing with the last forty years of the 19th century at a time when I had my first taste of computer usage and simple program development, I pondered on how much more effective it would be to have a program in which the student could place himself in the shoes of a North Dakota farmer sending his wheat to market in the days of railroad expansion and discrimination in long and short haul freight rates. Or how the student could more quickly understand the free silver issue of the 1890's if a simulation were developed wich a number of variables with roles for the student to play. True, the development of such games of simulation would take time in the short run, but in the long run, it appeared it would pay off in increased student understanding and provide a less cumbersome method of explaining these problems. A program of this sort is still waiting to be developed.

What I did develop was a sample module for a combination of tutorial and drill and practice program. It also occurred to me that one of the problems I had in teaching students with good native ability but poor educational background lay in my being certain that they were acquainted with the basic terminology and leaders of the period under discussion. These results showed that students often lacked such an understanding and were unable to use these terms correctly. They frequently were unable to identify the importance of a key figure. Data from the reading laboratory showed that the students often had a higher level of attainment in reading comprehension than in vocabulary. They often lacked confidence in their own ability. The program thus focused mainly on the cognitive domain. A minimum of role playing was included since the student was asked to place himself in the context of a given historical era in order to answer the questions. The computer was viewed here mainly as an adjunct to instruction, but it was also as a information delivery system. The program itself is written in BASIC-PLUS. Each segment of the program accesses 3 files with 3 different arrays, each containing 3 sequences of questions on 10 items. A sample of the program follows.



```
 1. IF YOU WERE A CARPETBAGGER, YOU WOULD BE
          (A) A POOR WHITE SOUTHERN FARMER
          (B) A NEW YORKER WHO CAME SOUTH TO HELP THE FREEDMEN
          (C) A BLACK ELECTED TO THE GEORGIA LEGISLATURE
          (D) A SOUTH CAROLINIAN WHO COOPERATED WITH BLACKS TO HELP THEM
          (E) AN OHIOAN WHO CAME SOUTH TO MAKE MONEY HELPING REBUILD THE
                   ECONOMY
  2  CORRECT ANSWER(S)
 ? A E

THE CORRECT ANSWER IS B E
LET'S CHECKON THE DEFINITION

1. CARPETBAGGERS WERE
         (A) NORTHERNERS WHO CAME SOUTH TO HELP THE FREEDMEN
         (B) SOUTHERNERS WHO COOPERATED WITH BLACK PEOPLE
         (C) SOUTHERN POLITICIANS
         (D) EX-CONFEDERATE ARMY OFFICERS
 1  CORRECT ANSWER(S)
? C
THE RIGHT ANSWER WAS A

CARPETBAGGERS WERE NOT POPULAR IN THE SOUTH. THEY WERE NORTHERNERS WHO
CAME SOUTH FOR A VARIETY OF REASONS, EITHER TO HELP FREEDMEN, OR
TO MAKE MONEY HELPING REBUILD THE RUINED SOUTHERN ECONOMY, OR TO WORK
FOR THE GOVERNMENT, EITHER IN THE ARMY OR AS CIVILIAN WORKERS.
LET'S TRY ANOTHER APPLICATION OF THE TERM

1. CARPETBAGGERS
         (A) WERE OFTEN WEALTHY PLANTATION OWNERS WHI DIVIDED UP THEIR
                  LAND WITH FREEDMEN
         (B) MIGHT BE EMPLOYEES OF THE U.S. GOVERNMENT
         (C) WERE OFTEN SHARECROPPERS
         (D) CAME FROM THE NORTH TO WORK IN THE SOUTH  FOR A VARIETY OF
                  REASONS
         (E) WERE POOR  WHITE TENANT FARMERS
  2  CORRECT ANSWER(S)
 ? E
 THE CORRECT ANSWER IS B D
 YOU SEEM TO BE HAVING PROBLEMS WITH THE TERM>
  PLEASE SEE YOUR INSTRUCTION FOR FURTHER HELP
 COME BACK LATER.[illegible]
```

A number of simulation programs as well as inquiry programs have been developed at the University of Michigan, Flint by Robert G. Shafer.[11] These programs basically utilize the games principle. They include a historical variation of the Ten Questions dealing with the Renanissance, and another (quite appropriately) deals with Jeremy Bentham's Felicific Calculus and its principles of decision making. Another deals with Roman history, creating a mythical battle between Roman and German armies giving the student an opportunity to devise strategy. Another program "Tycoons" enables the student to become the general manager of a 17th century English iron making concern and to make decisions on the basis of actual information used by the 17th century manager of the firm.

John Allswang of UCLA has developed a simulation program entitled "Little Italy". It enables students to palce themselves in the shoes of an Italian immigrant in the early 20th century and to make the choices the immigrant had to make. The program gives the student feedback on the probable effects of the choices they make. The simulation also takes luck into account since students make some choices at random and do not know what the effects were until the computer prints out the information. Charles M. Dollar of Oklahoma State University has developed a program, GHETT, which simulates life in the ghetto. In this simulation, probabilities determine the success of the student's attempts to get ahead. In general, simulation programs do provide a way in which students can "Get inside" a past era, can "get inside" someone else's life, thus providing something akin to a historical laboratory.

With the exception of one of Shafer's programs and Dollar's GHETT, existing indexes to computer programs show few CAI programs in history. The two major guides, one by Lekan and the other by Hoye and Wang list 18 separate programs written by 10 different authors.[12] Much of the work was done at two institutions: Carnegie-Mellon, the source of much innovation in history, and Oklahoma State University. Not mentioned in these guides are the computer based tutorial programs of over 500 questions and present information fundamental to the introductory course. The early American history programs consists of 20 programs of over 200 questions.[13] Among other programs developed, Betty Jehn of the University of Dayton has a version of Ten Questions for black history. At the University of Iowa, programs have been developed using census data enabling the students to raise questions about the Turner thesis. In a program of this type the computer stored data gives students enough data to be able to challenge conclusions of major historians. Another program developed at Ohio State by John H. Timmis III deals with historiography. Part of the course deals with events which have been computer programmed. The student is consequently able to control a large number of variables which interact with each other as well as a large data base.[14]

It is heartening to note that the Inter-University Consortium for Political and Social Research (ICPSR) at Ann Arbor has begun to develop material dealing with historical questions. ICPSR has begun to distribute subsets of larger data sets with a limited number of variables. These instructional subsets include 6 historical topics that are presently in an early stage of development. The titles include such topics as Unemployment in the 1930's, slavery in the U.S., and Congress during the first Hundred Days of the New Deal.[15]

Does CAI produce better teaching of history? The data is probably inconclusive. Preliminary surveys of the experience with the Notre Dame tutorials indicate the 98% of the students beleived that the CMI tutorials enabled them to perform better on tests, 78% believed they had learned more this way than from discussion groups, and 96% were willing to recommend the course to their friends. It is interesting to note that only 9% believed that the tutorials were dehumanizing and impersonal.[16] On the other hand, results of a study of the use of the STEUPS module on Voting Behavior in political science are more neutral. The results show that students did become more familiar with methodology and computer techniques through the use of these computer programs, and they became more familiar with the analysis quanitative data that has become more important both current and historical analysis of politics. Evidence did not show that they learned political science any better than in the traditional lecture format. But they did have a more positive attitude towards the course.[17]

But if the learning experience under CAI is better than the way the instructor taught it before, if the process is less ambivalent and more efficient, if teaching flows more directly out of its purposes, then CAI programs are useful. If the learning process in history is made more meaningful only for some students who find the interaction with the computer suited to their personal learning styles, it is worth the effort, for these students who might otherwise be turned off have become more directly involved with the problems of history. Generally, the studies that have been written indicate that greater student involvement does seem to be the case.[18]

In conclusion, what is needed to produce greater interest in CAI among historians? Beyond allaying the fears and the common reactions already discussed, it is hard to predict what will change attitudes. Certainly an openness to trying new techniques is necessary with the accompanying enrichment of learning strategies that comes from working up a computer program. The handful of historians working at CAI need at least a neutral attitude among their colleagues, more hopefully, one of encouragement.

This discussion has alluded frequently to the time element. Clearly, historians need released time to work on these projects. With this comes the need for funding, either internal or external. The funding bias seems to favor the sciences above the humanities. Whether this is because of the lack of interest or lack of lobbying strength on thepart of humanists (in which I include historians) is hard to say. A need exists. This positive attitude toward the effort to develop CAI history programs should also be exhibited by those working in the sciences and engineering where the use of the computer seems more "natural". One does occasionally run into the attitude that the computer and the humanities "don't compute." Humanists may sometimes stumble in learning computer languages because of their apparently mathematical structure and because of the fact that many text give mathematical examples. These should ideally be more cooperation and contact between historians working on computer programs. There is a need for programs that are already available and what is in the process of being developed. Hopefully in the area of funding, projects similar to the SETUPS (Supplementary Empirical Teaching Units in Political Science) sponsored by the American Political Science Association with NSF support could be developed for history.
One would hope that historians would not become so immersed in the past that they would fail to perceive the usefulness of CAI in playing a well thought out and well designed role in the learning process in history.

Notes.

1. Shafer, Robert G. "Computer in the (History) Classroom", AHA Newsletter, Vol. 15, No. 7, 9-10.
2. Rockart, John and Scott Morton, S., Computers and the Learning Process in Higher Education, New York, 1975. 236-241.
3. Burns, Robert and Pycior, S. "Computer Managed Tutorials in History", Unpublished paper, 1976. 5.
4. Burns and Pycior, 3.
5. Lower, Stephen E. "Making C.A.I. Make a Difference in College Teaching", ERIC Document, ED 129 299, 9.
6. Rockart and Scott Morton, 11-27.
7. Lower, 9-10.
8. Visich, M. and Braun, L. "The Use of Computer Simulations in High School Curricula." ERIC Document ED 089 740, Jan. 1974, 7-8.
9. Visich and Braun, 10.
10. "BASIC Simulation Programs: Vols. V and VI, Social Studies, 1971, ERIC Document ED 052 609
11. Shafer, 9-10.
12. Lekan, H. ed., Index to Computer Assisted Instruction, 3rd ed. New York: 1971, 142-144; Hoye, R.E. and Wnag, A., Index to Computer Based Learning. Englewood Cliffs, N.J.: 1974, 293-295.
13. Burns, R.E. "Freeing the Teacher to Teach: Computer Tutorials in History", Change Magazine Report on Teaching I, 1976, 42.
14. Timmis, J.H. III, "Syllabus for INCO 7_____: Historiography: The Philosophy and Methodology of History and the Place of Historical Methods in Communications Research." ERIC Document ED 125 042, 1975.
15. Stuckey, John, "Computer Applications in Teaching Political Science, Sociology and History," ICPSR. Unpublished paper, 1977.
16. Burns and Pycior, 3.
17. Broh, C.A., "Achievement and Attitude with Computer Related Instruction: A Field Experiment." ERIC Document ED 110 399, 1975, 14-16.
18. Whittenburg J.P., "The Computer as a Teaching Aid," The History Teacher, 1976, Vol. 9, 433-42.

A MICRO- AND MAXI- COMPUTER NETWORK FOR MEDICAL DATA BASE AND EDUCATIONAL ACTIVITIES

David P. Yens, Ph.D.
Medical College of Georgia and
Georgia Joint Board of Family Practice

A network of stand-alone microcomputer systems linked to a large central mainframe is being developed to provide a resident information system as a solution to the following needs by family practice resident training programs in Georgia: 1) maintaining a record of clinical experiences; 2) assessing the quality of clinical performance; and 3) insuring adequate experience with a variety of patient problems. Minidisks are used for temporary local storage to minimize communications costs. The microcomputers will also contain computer based patient management problems and administrative and patient accounting programs. The system will greatly benefit the resident education programs and will provide good examples of the ways computers might be used in physician's offices. Generalizations of the model are made to public school settings.

Background

The training of medical physicians is a complex process that requires a number of years. First, four years are spent in medical school to obtain an M.D. degree. However, the M.D. is not sufficient to practice medicine; one to seven years of supervised clinical work is required before a Doctor can practice medicine, depending upon the specialty selected. This postdoctoral work is usually called residency training; the resident is involved in extensive clinical activities, structured discussions about patients that have been seen, lectures and conferences, and other experiences that provide an in-depth knowledge of the principles and practice of medicine.

In the Family Practice specialty of medicine, a three year residency is usually required. This experience involves time spent working in other specialties of medicine (internal medicine, pediatrics, obstetrics/gynecology, etc.) and a significant amount of time caring for families on a continuous basis. The hallmarks of modern Family Medicine are continuity of care and the provision of comprehensive health care. The comprehensive care includes maintenance of both physical and mental health and also involves managing patients who may be referred to other specialties for treatment. Thus, the family practitioner must be a very well-rounded individual with broad knowledge of many aspects of medicine who is devoted to providing long-term medical care for families, both physiologically and psychologically. However, this very breadth of knowledge generates some unique problems in the training of these residents.

Problems

Three common problems in the training of family practice residents are:

(1) Obtaining and maintaining a record of the clinical experiences that each resident encounters;

(2) Developing and using an accurate assessment of clinical performance;

(3) Insuring that there is adequate experience with a variety of patient problems.

A record of clinical experiences includes the specific kinds of medical problems that the resident has seen and treated. This list is required for two purposes: 1) to give faculty the opportunity to review the kinds of experiences the resident has had to insure that he or she has performed a sufficient number of medical procedures to demonstrate proficiency and so that an appropriate balance of exposure to medical problems may be maintained; and 2) to assist the graduated resident to obtain hospital privileges. At the completion of the residency program, the physician must typically seek hospital privileges in the area where he practices. One of the types of information required for obtaining these privileges is a list of medical experiences and problems the resident handled during the residency. The record of clinical experience constitutes that document.

At present, the list of clinical experiences is maintained manually in a somewhat haphazard manner. The clerical problems inherent in maintaining an accurate list are quite difficult when a number of residents are involved in a clinical training program. Typically, the resident is so busy practicing medicine that he or she does not keep accurate records.

The assessment of clinical performance is typically done through subjective evaluation of observational and patient record data by medical faculty who work with residents. Due to the normal heavy load on these faculty, it becomes very difficult to maintain an accurate perspective on a given resident throughout the three years of the resident training program. Furthermore, a given resident may be observed and/or evaluated by several faculty. Each faculty member involved with the assessment process will have his or her own criteria for adequate performance, and may not incorporate a broad view of all the specifics concerned with patient care and treatment.

During a resident training program, a resident in family practice medicine will tend to be exposed to the specific patient diseases and problems that are prevalant in the area where the residency is performed. Thus, there may be a large number of important problems that may be seen rarely, if at all, and there are likely to be gaps in the resident's experience with medical problems.

In addition to the above three problems, current residents in family practice are practicing in the middle of the computer age. Unfortunately, very few medical schools incorporate any familiarity with computers into the medical curriculum. However, as we all know, the computer is being used in an increasing number of areas in medicine, including extensive patient records keeping in hospitals, computer assisted tomography (CAT scans), computer assisted evaluation of EKG's, computerized patient billing and accounting systems that are marketed by private firms, and several other areas. There is no question that residents in medicine today will receive increasing exposure to computers in many areas of medicine tomorrow. The earlier that they become exposed to computers, the more effectively they will be able to use computer technology when they encounter it.

Computer Systems as a Solution

For problems involving the large-scale storage and retrieval of data, a computer system is an obvious solution. A well-designed information collection, storage, and retrieval system can process the data necessary for providing the record of clinical experiences; for providing some assessment of clinical performance; for providing the balance of experience needed and, in cases where exposure to certain kinds of cases is unlikely or an insufficient number are available, provide experience with these kinds of cases through computer-based patient management problems. Furthermore, the computer system used for these purposes can be used for business and accounting activities and as a medium for familiarizing residents with the capabilities of computers.

The development of such a system is being supported by the Georgia Joint Board of Family Practice. This computer-based resident information system (CRIS) is available in all five Family Practice resident training centers in the state of Georgia. CRIS supports the collection, storage, and retrieval of data concerning all clinical experiences by residents in family practice, provides the opportunity for interactive instructional experiences using patient management problems, and permits the early familiarization by the residents with the potentials of computer technology. An additional aspect of this system is that it also permits the development of programs for patient billing and accounting, patient recall, physician scheduling, and other programs that are of value to both residents and administrators.

The family practice resident training programs in Georgia are located in all parts of the state, from Savannah to Columbus to Rome to Augusta. Because of the need for long term mass storage of data and because of a desire to be able to access simultaneously the data from all resident training programs for research on statewide medical activity, it was decided that a large central computer should be used for the major storage and retrieval operations. At the same time, the geographical distribution of the centers, the nature and extent of the data recording activities, and the desire to minimize transmission costs and the potential for service interruptions led to a decision to obtain a stand alone desk top computer and a printing terminal for each of the resident training programs.

System Configuration

This desk top computer, an Intelligent Systems Corp. Intecolor 8051, is equipped with a large memory (47 k total with 24K program-usable bytes), minidisk drives with an 80,000 character capacity per disk for local data storage, a powerful BASIC language, and an 8-color 19 inch display screen that is configured for 48 lines of 80 characters each. With this system, the clinical experience data may be entered locally via a formatted entry and stored on a disk until a convenient time at which all this data can be transmitted in a batch to the central computer. In addition, the stand alone system permits local operation of administrative and educational programs without the need to tie in to a central computer. With this configuration all the resources of a large scale computer are available when needed, but it is possible to do a significant amount of work locally without connecting to the central computer. It is anticipated that this system will provide better reliability, simpler operation, lower total cost, and faster operation than those systems which use CRT terminals connected by telephone lines to a central computer, and more flexibility than systems where all transactions and activities are done on a small local computer.

The State of Georgia has a unique facility to handle this kind of system. The Georgia University System Computer Network serves all state-related institutions of higher education through a sophicated network of time sharing lines and multiplex nodes. The Cyber 70/74 computer located at the University of Georgia in Athens has more than ample capacity to handle the mass processing requirements of the project. A sophicated data base management system, DMS-170, is available on the computer for data storage and retrieval.

The ISC 8051 systems can be used as terminals for communication with the Cyber as well as stand alone computers. The terminals are equipped with field protection and graphics capabilities so that sophicated forms input may be used and tabular or graphical output of data may be displayed.

It is believed that the system described above constitutes a powerful model which can be incorporated into a private physician's office no matter where it might be located.

Data Input

Each time the resident interacts with a patient, whether this interaction occurs in the clinic, on the telephone, or in the hospital, the resident is required to complete a form, called an "encounter form," including the patient number, time, date, and place of encounter, type of contact, participating personnel, reported problem, actions taken, and other relevant information. These forms are checked by the faculty member responsible for the activity and are then sent to a data entry clerk for entry into the computer system.

Because different kinds of encounters result in different kinds of information, different formatted screens are provided for each kind of encounter. For instance, a data base encounter is designed to provide a patient history and background; this information is much more extensive than the other kinds of encounters and provision is made for this by providing for the entry of birthdate, number of people in family, specific history information for females, and other relevant data. A telephone encounter is quite limited and does not make provision for immunizations or physical findings. Prior to input, a code book is used to establish the appropriate codes for the medical problem, personnel, procedures, tests, treatments, and results.

The encounter record for a normal office visit provides for the input of the following data: patient number; date, time, and place of contact; type of contact; codes for the medical personnel associated with the contact, the problem, procedures used, diagnosis made, immunizations and tests conducted, prescribed treatments, referrals, results or actions, and outcomes; return appointment date and time, financial information, and in the case of a patient history (data base) record, appropriate personal and demographic data. From this list, it can be seen that this is not a complete patient record as might be gathered in a hospital. Rather, it is designed to provide basic information concerning the providers involved and the kinds of activities that they conducted. The additional information is collected for research purposes and to permit the comparison of activities that occur for the same problems by the same resident at different times and/or by different residents at one program or across programs. Security of the file is maintained through password access to the program and the file and through encoding patient numbers at the site of entry.

A special problem resulting from working with different family resident training programs is that each is presently using a different encounter form. The desk top computer permits us to develop a data input form which is tailored to the specific program but which converts the data to a common form for final storage in the central computer system. At present, also, not all information is collected by all programs; each program has already developed its own encounter form to meet local needs. At present, it is not appropriate to require all programs to adopt a common form, but as experience is gained with the system it is anticipated that each program will start collecting all the desirable information and will move toward accepting a common form.

The data entry program checks for appropriate ranges and kinds of data that are entered in order to trap any obvious errors before they get into the data base, and the data entry clerk re-enters data that is flagged as erroneous. Thus, every effort has been made to make the entry of data into the desk top computer as simple and accurate as possible.

The data for one or more days is stored on the 8051's minidisks until it is transmitted in a block to the central Cyber computer. The transmission takes only ten to twenty minutes at 300 baud. Because of this, it would be feasible to perform this activity after 5 p.m. when telephone rates are low. Thus, this use of storage on minidisks should result in a very low communications cost. A program in the Cyber checks and validates the data as it is entered; any errors result in non-storage of the record and its placement, with an indication of the error, in a message file which is returned to the terminal upon completion of the data transmission. It is then possible to check and update any records in which an error was made. Because the data is held on disk for a while, it is also possible to update records with test results and other data received after initial entry. The data may be held on the disk for a period of time to provide a back up in the event that something happens to the information on the large system.

The validity checking at the terminal and by the central computer is designed to minimize the number of errors that are made in the data base. Effective update procedures are available for modifying data if this is required; these are performed on-line with the Cyber using the Query Update system.

Although limited retrieval programs are available for retrieving data from the local disks, the Query Update program permits flexible and sophicated retrieval and reporting of data stored on the Cyber. Query Update permits sorting, summing, frequency tabulation, conditional selection, and other powerful activities. All of these facilities will be used to create reports that will assist with the solution of the problems described at the beginning of this paper. The reports will be printed on the printer attached to each Intecolor 8051 System. In the future, it is planned to develop individualized report formats for each of the family practice programs.

Outputs that Help Solve the Problems

Problems 1 and 3. The printed tabulation of all medical problems seen by residents will permit an analysis of the residents experiences and will perimt the adjustment of resident assignments so that they might receive an adequate balance of experiences. The desired balance will be determined by the faculty of each program. The resident experience list will be compared with the desired experience pattern and deficiencies noted. This will occur every six months to provide for adequate longitudinal planning of experiences. As mentioned above, this is especially important in that the family physician is expected to be able to handle a wide variety of problems in medicine. An example of this tabulation is given in Table I. At this time only problem codes are used; in the future, a translation table will be developed to provide the problem names.

TABLE 1

Sample Output from the
Summary of Resident Problems

RESIDENT NUMBER: 012
LOCATION NUMBER: 03
DATE OF RUN: 04/08/78

PROBLEM CODE	PROBLEM DESCRIPTION*	FREQUENCY
033	Whooping Cough	2
055	Measles	23
201	Hodgkin's Disease	1
	:	
y89	Family Relationship	4

* The description will not be available for the first version of the programs; appropriate tables will be developed and this feature will be added later.

Problem 2. The evaluation of the quality of health care can be obtained through a date-sorted listing of all problems of the same kind that were seen by a given resident along with the other data such as other reported problems, procedures used, diagnosis, tests, treatments, whether referral was made, results and outcomes, and when return appointments were specified. An example of this output is given in Table 2. In the future, descriptive attributes will replace code numbers. At present, it is felt than an interoccular evaluation of activities associated with a set of cases will provide a good indication of the quality of health care provided. An experienced physician faculty member will determine whether the diagnosis, treatments, etc. were appropriate to the problem and provide additional instruction where needed. These data may be used to evaluate whether progress is made over a period of time and to identify specific areas in a resident's treatment process that needs special attention. However, this is only one dimension of quality, and other factors such as interpersonal relationships and the quality of the dictation must also be evaluated to assess the complete quality of the care.

Problem 3. The process of insuring that residents receive an adequate variety of experiences with different kinds of patients

Table 2

Sample Output for Health Care Quality Evaluation

RESIDENT NUMBER: 012
LOCATION NUMBER: 03
DATE OF RUN: 04/08/79

PROBLEM NUMBER	DATE OF INTERVIEW	OTHER PROBLEMS	DIAGNOSIS	PROCEDURES	TESTS	TREATMENTS	REFERRALS	RESULTS	OUTCOMES	RETURN DATE
033	06/09/78	-	033	3580-017	3580-005	3551-015	-	0	0	06/20/78
033	06/20/78	-	033	-	-	-	-	2	2	-
055	08/12/78	034	055 034	-	3551-030	3551-021	-	0	0	08/25/78
055	08/26/78	034	-	-	3551-030	-	-	2	2	-
.										
.										
.										
201	02/22/79	-	201	-	-	-	022	3	-	03/12/79

was described above. In those situations where certain kinds of experiences are desirable but the patient population does not contain the desired kinds of problems, patient management problems may be assigned. Problems of this type have been found effective for physician training at several medical schools; problems appropriate for family practice are currently under development. It is anticipated that these problems will be administered by the Intecolor 8051 on a stand-alone basis, with each problem residing on a different minidisk.

The problems will be as sophicated as possible given the capabilities of the system, and will involve extensive branching logic and evaluation capabilities to make the problem as realistic as possible and to provide useful feedback to the user. Note that these problems should be as valuable to faculty and other health practitioners as they will be for residents, and should see widespread use. Discussions have been held with the various directors of family practice training programs to determine the kinds of problems with which residents should have good familiarity but with which they would probably receive minimum experience in the area where they are presently located.

Residents will be able to work on these problems when on call in the center or during the early morning and evening hours. They should be of value to family practice faculty, medical students, and other health personnel in addition to their importance for residents.

The research possibilities of the proposed data base are impressive. Initially, it is of great interest to determine the frequencies of medical problems that are being seen by family physicians at all these centers combined. These frequencies will help to define the content of family practice within Georgia. However, it will be possible to answer a number of other questions such as patient load, patterns of appointments, the amount of time spent with patients, frequency of return visits that are made, and uniformity of care over several providers furnished for each of a variety of problems seen.

Patient accounting, billing, and other business-oriented activities will be made available on the systems primarily to demonstrate to residents how a small computer could be used for that purpose. They may also be used in parallel with manual systems to verify accuracy and, in some cases, replace manual systems.Until computers become common in physicians' offices, however, most programs prefer to use manual systems for the resident training activity. One program that will be heavily used is a patient recall system which will permit the listing of all recall appointments for a specific week or month to facilitate contacting patients and scheduling appointments. It is anticipated that this recall system will be able to improve patient scheduling activities and enhance the continuity of health care.

Application of the Model in Other Settings

The model described above may be appropriate for any network in which lengthy time sharing interactions between remote sites and a central computer are necessary for transmitting and storing data, where significant communications costs are a consideration, and where multiple system uses are anticipated that could be performed on a powerful microcomputer configuration. A good example of this kind of need might be found in a small but geographically large school district or consortium of school districts.

Bukoski and Korotkin (1975) documented the growth of computer uses in secondary education; some 58.2 percent of schools who responded to their survey reported use of computers for administrative and/or instructional purposes and the trend was toward using the computer more fully. They predicted that at present levels of growth in use every secondary school in the country will have access to a computer system within the next decade. Use of the computer as a "Problem Solving Tool" and as a subject area for "Computer Science" courses were the most frequent applications; these applications were expected to remain dominant for the next decade. However, they point to the availability of funds as a major barrier to growth.

Microcomputer systems furnish a good approach for meeting these secondary school needs at a low cost; the increasing number of these systems that are being obtained for secondary schools indicates that teachers and administrators are recognizing this. These systems can form the basis for a powerful and effective instructional and administrative network using the micro-maxi-network model, especially if there is a large distance between the schools and the district computing center. This concept is further enhanced by the recent appearance of microcomputer systems that can support several terminals. A network based on microcomputer systems at each school would need a smaller mainframe and fewer timesharing ports than would be required to support all administrative and educational functions on one terminal-based timesharing network, and the savings in communication costs would probably pay for the amortization of the micro systems over five years. Thus, a school system could "have its cake and eat it too."

The Future

The basic elements of the system described above are in the final stage of development; the administrative and patient accounting applications will be developed during the next two years. Patient management problems will continue to be developed. It is expected that much will be learned from this initial configuration and that many changes will be made. However, this is seen as the basis for a large-scale implementation that may include several medical specialties as well as private physicians and a system that will include sophicated inter-user communications. The primary focus for private physicians is on the rural practitioner who may feel that he or she has little contact with the "outside world." If a computer system can be developed that brings the world to the rural practitioner, it may be possible to encourage more physicians to move to these locations where they are critically needed, thus providing a solution to the physician distribution problem.

Summary

A combination of microcomputers and a large central mainframe for data base storage and management has been shown to be a viable solution to several problems facing family practice resident training programs. Although starting with five training sites, this is expected to be the basis for a large network encompassing several medical specialties and private physician practices which can significantly improve the training of residents and the management of private medical practices.

Reference

Bukoski, W.J. & Korotkin, A.L. Computing Activities in Secondary Education, Final Report. Washington, D.C.: American Institutes for Research, 1975

Acknowledgements

Partial support was provided by a General Research Support Grant from the Dean, School of Medicine, MCG, for the development of the initial data base system by the Research and Educational Support Section, Division of Systems and Computer Services, Medical College of Georgia.

Primary project support has been provided by the Georgia Joint Board of Family Practice.

COMPUTER-ASSISTED GUIDANCE

THE GUIDANCE INFORMATION SYSTEM:
A COMPUTER-BASED CAREER EDUCATION TOOL

Linda L. Kobylarz
Time Share Corporation

The Guidance Information System (GIS) is a computer-based interactive, data-retrieval system which provides access to information about occupations, four- and two-year colleges, graduate schools, and financial aids. GIS is a composite type of resource which brings together much of the data needed in the initial stages of the career decision-making process. This paper will discuss how the GIS addresses three of the key areas usually included in career education programs: the self-awareness area, knowledge of lifestyles and career roles, and the development of decision-making and planning skills.

Background

The concept of Career Education has been nationally advocated as a means to more fully meet the needs of those facing both the choices and demands presented by a rapidly changing world. In the broadest sense, Career Education is a developmental process through which individuals can be helped to gain knowledge of themselves and the world of work and to learn the skills needed for effective planning and active participation in the greater society. Three of the key components of most Career Education programs focus on: increasing a person's self-awareness, increasing his/her understanding of lifestyles and career roles, and increasing his/her skill level in decision-making and planning. Basic to each of these elements is the availability of sound information. Whether the information is about an individual's interests and abilities or about the potential job market for lawyers in 1985, it must be readily accessible, accurate, and up-to-date.

Much of the responsibility for providing Career Education related information falls within the realm of school Guidance personnel. Throughout the country, there are many attempts underway to implement new approaches to career guidance. That these new approaches are sorely needed has been highlighted by a recent study of the career development needs of high school students. (Prediger, et al, 1973, p. 9) The students, chosen from a nation-wide sample, indicated that help with making career plans was their most important area of need. This was in sharp contrast to the amount of help in career planning students said they received from their counselors, The students in the Prediger study also evidenced an alarming lack of knowledge about the world of work per se (Prediger, et al, 1973, pp. 29-33).

The importance of information to the career decision-making process is often cited in the relevant literature. In his well-known work on career development theory, Donald Super notes that information concerning occupations is a vital element of the crystallization and specification stages of vocational development (Osipow, 1968, p. 124). In the "Conceptual Framework for Occupational Choice" described by Blau, et al (Bailey and Stadt, 1973) occupational information is characterized as one of the immediate determinants of occupational choice. Hoppock sums up the role of occupational information in this way: "it...affects occupational choice by helping us to discover the occupations that may meet our needs and by helping us to anticipate how well satisfied we may hope to be in one occupation as compared with another" (Hoppock, 1967, p. 112).

MEETING THE NEED

Recognizing the critical need to provide career information to students, forward looking guidance departments are reassessing not only the quantity and quality of the information now available to their students, but also, the manner in which the information is presented. The utility of traditional methods of career information dissemination is being questioned and new techniques of accomplishing the objective are being sought. The Guidance Information System (GIS) draws upon the latest technological innovations to help counselors deal with the problem of providing a career information delivery system that is current, accurate, and easily accessible. The GIS is a computer-based system which provides instantaneous remote access to information about occupations, colleges, graduate schools, and sources of financial aid.

The Guidance Information System traces its roots to a project conducted at Harvard University in 1966 by Dr. David V. Tiedeman, under the sponsorship of the U.S. Office of Education. The GIS is now being maintained and further developed by the Education Division of Time Share Corporation, a wholly-owned subsidiary of Houghton Mifflin Company. The System is being used in 33 states, Canada and Great Britain in a wide variety of settings which include junior and senior high schools, junior colleges, community colleges, four-year colleges, adult career counseling centers, regional career education resource centers, CETA centers and correctional institutions.

The actual operation of the GIS is really quite simple. A user "communicates" with the computer storing the GIS information through the terminal which allow him/her to interact in a unique way with the desired information. Individuals working with the GIS use the GIS Guide, which details the characteristics about occupations or colleges, to select those items they might want to include in their search. The GIS user can quickly get a list of occupations or colleges that meet specific criteria and detailed information about any item on the list. The GIS Guide is structured in such a way that it provides initial broad organization to a search, while at the same time allowing for great flexibility within that organization.

THE GIS INFORMATION FILES

The Occupational File contains detailed information on approximately 1,200 occupations and references another 3,000 occupations. A user can explore the information through the 15 Office of Education clusters and the DOT job families. Individuals can also choose from characteristics of occupations such as aptitudes and interests usually related to an occupation, education levels required, salary ranges, etc. An important feature of the Occupational File is a cross-referencing system that directs users to a wide variety of multi-media materials for more information about occupations. The information found in this file is gathered from Federal and State agencies, trade unions, professional organizations, and businesses throughout the country. The information in the Occupational File, as in all other GIS files, is updated twice a year. The Armed Services Occupations Information Section of the Occupational File contains basic information on more than 100 occupations in the Armed Services. For each occupation there is a general description, a list of related military occupations, a list of related civilian occupations, and the average length of training.

The Four-Year and Two-Year College Files enable users to explore information on more than 2,700 colleges across the country. The files contain information on college program offerings, costs, campus life, national test scores, special programs, etc. GIS users may examine a list of over 300 characteristics in making choices about college selection. The information found in the college files comes directly from the participating colleges.

The Graduate School File contains information on hundreds of graduate schools and their programs. Information is available about degree requirements, placement, financial aid, etc. The information found in the GRAD File comes directly from the participating schools.

The Financial Aids Information File details financial aid possibilities from sources such as Federal and State Governments, foundations, businesses, the military, trade and labor organizations, and religious and charitable groups.

THE GUIDANCE INFORMATION SYSTEM AND CAREER ED

In the "Self Awareness" component of Career Ed, the focus is on helping individuals understand themselves and others. Individuals increase awareness of their own interests, aspirations, aptitudes, abilities, and values. The Occupational File of the GIS encourages users to consider their interests and aptitudes when exploring occupations. In the GIS, occupations are grouped according to the kinds of activities in which workers are involved most of the time. The work activities fall into three broad categories:

- activities that involve working with people most of the time
- activities that involve working with ideas or data most of the time
- activities that involve working with things or machines most of the time

The GIS Guide helps users to focus on those work activities which most closely reflect their own interests. The GIS is also cross-referenced to several off-line interest inventories. The GIS Instruction Manual helps those who have taken an interest inventory, such as Holland's "Self-Directed Search", to easily relate the results of that survey to characteristics found in the GIS.

The aptitude section of the GIS relates various aptitudes (verbal, numerical, motor coordination, etc.) to the skills and abilities workers need to do their jobs in a satisfactory way. GIS users can develop a list of occupations which require certain aptitudes and at the same time, they can instruct the computer to leave out occupations that require certain other aptitudes.

The Career Ed area of "Understanding Lifestyles and Career Roles" emphasizes the concept of job clustering and job families and how various aspects of a job can affect a person's lifestyle. Two occupational clustering systems are used in the GIS: The U.S. Office of Education system of fifteen clusters and the Dictionary of Occupational Titles (DOT) job families system. GIS users can also use educational levels to group occupations. Both the Office of Education clusters and the DOT families can be easily related to academic areas thereby reinforcing another Career Ed theme; helping individuals to more clearly see the relationship between academic areas and occupations.

The GIS also provides information on lifestyle ramifications of specific occupations. Information on working conditions, work hours (evenings, over-time), whether or not travel is involved, earning ranges, occupational outlook, etc. help a person to understand more fully the impact an occupation might have on his/her lifestyle.

REFERENCES

Bailey, L., Stadt, R. *Career Education*. Bloomington: McKnight, 1973.

Hoppock, R. *Occupational Information*. New York: McGraw Hill, 1967.

Osipow, R. *Theories of Career Development*. New York: Sippleton-Century-Crofts, 1968.

Prediger, D. et al. *Nationwide Study of Student Career Development: Summary of Results*. Research Report No. 61, Iowa City, Iowa: The American College Testing Program, 1973.

THE ROLE OF THE COUNSELOR IN RELATION TO COMPUTER-ASSISTED CAREER GUIDANCE

James P. Sampson, Jr.
Student Counseling Center
Georgia Institute of Technology

One of the most important applications of technology in counseling and guidance in recent years has been the use of computers in career guidance programs. Research shows that certain limited career guidance functions can be performed by the computer equally as well, or better than by the counselor. Given that computers are effective in delivering certain services, what is the optimal role for the counselor in a computer-assisted career guidance program. This study provides suggestions for specific counselor intervention strategies to facilitate student career development.

The integration of two important advancements of this century, career guidance and computer technology, has resulted in the development of computer-assisted career guidance systems. The objective has been to utilize the computer for storing, manipulating, and retrieving information, thus freeing the counselor to assist the student in integrating this information with psychological factors such as self-concept, needs, and personality, which influence career choice.

A number of studies have been conducted to compare the effectiveness of counseling versus computer interaction in relation to identified career development outcomes. Results show that certain limited career guidance functions can be performed by the computer equally as well, or better than by the counselor. However, many questions concerning the conditions for the most effective utilization of both counselors and computers remain. What should be the counselor's role in assisting students to obtain maximum benefit from computer-assisted systems? What degrees of counselor intervention produce the best results?

The effects of three degrees of counselor intervention on (1) the community college student's perceptions of level of career development, (2) the evaluations of the System of Interactive Guidance and Information (SIGI), and (3) the way the SIGI system was used were evaluated by employing a quasi-experimental pretest-posttest research design. A total of 124 community college students from three groups, structured counselor intervention, non-structured counselor intervention, and no counselor intervention (control group) participated in the study. Subjects for the structured and non-structured groups were drawn from eight sections of a behavioral science course. The subjects from the control group were drawn from students who used SIGI on a walk-in basis.

The nature of the counselor intervention in the structured group was guided by the assumption that students will profit from counselor intervention strategies that are planned to follow the students' use of the computer system. The specific intervention strategies were: (1) orientation to the SIGI system, (2) introduction to each of the six subsystems, (3) discussion of the students progress through the system, including problems encountered at the terminal, (4) referral to other appropriate career and academic resources, and (5) individual counseling initiated at the student's request. The findings of this study suggest that this assumption is valid when student evaluations of their experience are taken into account.

The nature of the counselor intervention in the non-structured group was guided by the assumption that students would profit from having the freedom to utilize counselor intervention strategies as they desired. Specific intervention strategies, such as those used in the structured group, were not introduced by the counselor/instructor. This is not to assume that those strategies were not used. Students could at any time request assistance from the counselor/instructor, thus initiating a particular intervention strategy. The findings of this study suggest that this assumption is not valid.

The absence of counselor intervention in the control group was guided by the assumption that SIGI provides a comprehensive approach to career exploration and decision making that can be utilized without other support strategies. The findings of this study suggest that this assumption is valid when student evaluations of their experience are taken into account. Despite the absence of counselor intervention in the control group during the use of SIGI, 50% of the students had scheduled, or planned to schedule, an appointment with a counselor. This suggests that many students who do not receive any counselor intervention during their use of a computer-assisted career guidance system request some type of counselor intervention at a later time.

Two implications are suggested by this study. First, when computer-assisted guidance programs are used as a part of a course encompassing career development, an approach utilizing structured counselor intervention strategies is more advisable than an approach using non-structured counselor interventions. Second, computer-assisted career guidance systems should be made available for those students who wish to use such systems without counselor intervention. However, counselors should be available when students encounter problems at the terminal and when they desire counseling after the system is completed.

References

Sampson, Jr, J.P. Counselor intervention with computer-assisted career guidance (Doctoral dissertation, University of Florida 1977). *Dissertation Abstracts International*, 1977, *38*, 3297A (University Microfilms No 77-25,952)

COMPUTER-ASSISTED TESTING

AN AUTOMATED SYSTEM FOR GRADING BASIC PROGRAMS

Donald R. Chand
Department of Information Systems
Georgia State University
Atlanta, Georgia 30303

Abstract

This paper describes a system for computer evaluation of student BASIC programs and the experience of using the system in the teaching of a first course on data processing for business majors at Georgia State University. The practical advantages of such a system in the teaching and learning of programming are brought out in the paper.

1. Introduction

Drill and practice is a well accepted technique for learning any new material. It is, however, an essential process for learning the art and craft of computer programming. Therefore in any teaching of programming, students are required to do practical programming exercises which are graded by the instructor. Clearly such grading must first consider the formal correctness of the program and then the other desired attributes of good programs, such as readability, understandability, reliability and efficiency. Checking the correctness of a program is inherently a complex task and, as a consequence, grading of computer programs tends to be often subjective and inconsistent. In addition the repetitive and routine nature of grading makes this activity, in a large university environment, somewhat tedious, difficult and time consuming. The idea of using a computer to assist the instructor in this task of grading programs naturally arises.

The concept of a computer-controlled grading scheme is not new. We are well aware of the power and usefulness of computer grading programs that not only grade multiple choice examinations but produce valuable statistics on the class performance and the relative difficulty of the questions. Forsythe and Wirth [3], however, appear to be the first to report the development of a computer program that graded problems in programming. Their grader was designed to handle numerical analysis exercises in Algol. Temperly and Smith [4] describe a more general program for grading programming exercises written in PL/1. The success of these grading programs motivated us to initiate the development of a complete system for automating the tasks of grading computer programs. It should be emphasized that as opposed to these earlier efforts of creating a program for grading programming exercises, our goal was to develop a flexible and versatile grading system composed of several parts, one of its components being the automatic grader. Since BASIC is the programming language used in our required course IS 301, an introduction to data processing, we developed a system for grading BASIC programs. The fundamental objectives of the system are:

- It should relieve the instructor of the routine and repetitive aspect of grading
- It should better evaluate the correctness of programs
- It should make grading of computer programs more consistent and objective
- It should aid the process of learning the science of programming.

An automated grader system for evaluating BASIC programs has been designed and implemented. The system has been in use for three quarters and the informal reaction of both the students and faculty is generally positive. This paper describes the system from the users point of view and, therefore, does not contain a detail description of the architecture of the system. The emphasis is upon the functions and how the functions relate to each other. An attempt has been made to extract and

and characterize the properties and features that are essential for a grading system in order for it to be an effective aid in the teaching and learning of computer programming.

2. The Structure and Function of the Grading System

The automatic grading system is composed of several parts that interact with each other to collect, organize and produce information concerning the performance and correctness of students BASIC programs. The kernel of the system is a complete BASIC interpretive compiler, written in Algol W, which interprets and executes BASIC language instructions by reading input data from a predefined file. The system compares the results with data in the corresponding results file and the outcome is recorded on a disk file for instructor's use. The system also informs the student how well his program performed. From the point of view of the student the grader furnishes input data to his program and mysteriously evaluates whether the program is running correctly or not. We believe that an immediate response to student that his program is either correct, or is partially correct, or is incorrect helps the student to better learn the inherent complexity of designing correct programs and ways of dealing with this stringent requirement of programming. It is evident that for the grader to do a meaningful evaluation, the data file created must be designed carefully and must be comprehensive. We discovered that the data set must at least contain test cases for correctly evaluating all the distinct paths that can be invoked during the execution of the program.

Each time the grader is invoked it accumulates data that is deemed useful for assigning a numerical grade. Specifically it records the program size, the number of times this exercise has been graded by the system, a measure of program correctness in terms of the ratio of correct cases and total number of cases, the date the exercise was graded, and the execution time. Notice that the system merely records this information, the numeric grade is still assigned by the instructor. A report generation program exists that processes and prints this information in a form that makes grading more objective and consistent.

It is clear from our discussion so far that the most time consuming task for the instructor now is the creation of a comprehensive and complete data and results file. However this task needs to be done only once for each problem. A facility has been provided to add new problems to a library of exercises with predefined data and results file. The library concept has far reaching implications. In particular, it started a trend towards the standardization of the material covered in the various sections of the same course taught by different instructors. We present a bird's eye view of the system in figures (1) and (2). The functions and their interactions are captured in HIPO, Hierarcy plus Input-process-output, diagrams. The information required to use the system is presented to provide a clearer view of the system.

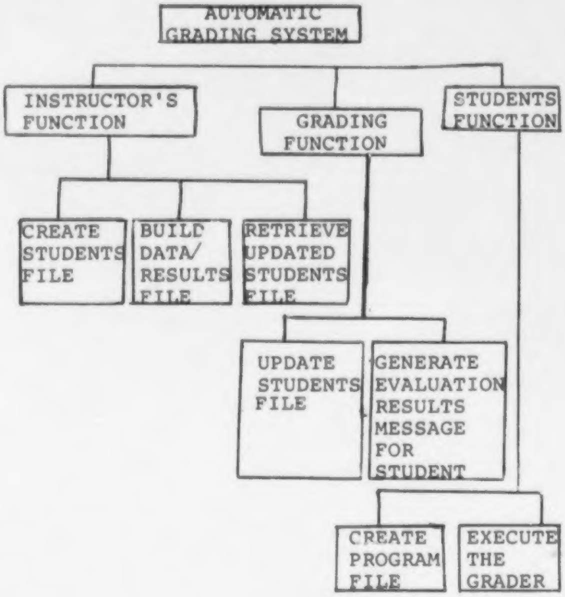


Figure (1): A functional model of the system

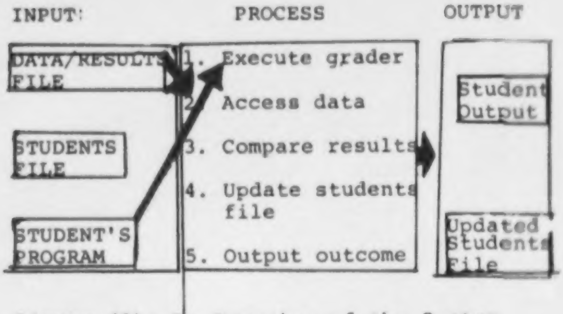


Figure (2): An Overview of the System in Use.

2.1 Creation of Data/Results File

The purpose of the data/results file is to hold the test data and the corresponding correct results which the grader needs for evaluating the correctness of student programs. On UNIVAC 7, under the interactive VMOS operating system, the instructor builds a SAM file in the following format:

line 1: Total Number of (input) values
lines 2-p: The list of data items
line p+1: Total number of result values
lines p+2...: Followed by a list of results

For obvious reasons the student is unaware of the existence of these files. Although each instructor may build such files on his own, we have chosen to coordinate this activity in such a way that a uniform library of problems and corresponding data/results files are maintained by the department. The practical advantage of this centralized coordination will be elaborated later.

2.2 Command for Using the Grader

One of the design goals of this system was that it should be easy to use. The grader is invoked by one simple command shown below. The interpreter has been designed to track common typographical errors and, in case of errors, the system requests the user to retype the command.

The syntax of the command is

```
/DØ  GRADER (<program name>, <problem
             number>, <student ID>,
             <student last name>)
```

where,

<program name>	identifies the specific program residing on the disk that is to be executed
<program number>	is used for identifying theappropriate data/ results file
<Student ID>	and <student last name> together identify the record in the student file that will be updated.

Students are encouraged and expected to first debug and test their programs using the UNIVAC BASIC compiler in the usual way. The GRADER should be invoked only when the student is reasonably confident that his program is functioning correctly. Each grader run is recorded and the number of runs may be an evaluation criterion.

2.3. Creation of Student Name File

The purpose of the student name file is to provide an area for the system to record data concerning the execution and performance of student program. The instructor creates an ISAM file containing the name of each student as it appears on the class roll. In this system the name of each student must appear on a new line in columns 1-20.

Each time the grader is invoked by a student, it collects certain facts concerning the program and its execution. At the end of execution, the system updates the following fields in the student name file.

Field 1: Number of times this program has been graded
Field 2: Ratio of correct solutions and total number of cases
Field 3: Execution time
Field 4: Program length
Field 5: Grading date

Students may run the grader as often as they desire to improve the performance of their program. Field 1 keeps track of the total number of grader runs for the problem. The remaining fields are self explanatory. A report generation program is available to output the student file in a form most suitable to the instructor.

2.4 An Illustration

Suppose the instructor assigns the following exercise to his class and desires to use the automatic grader for grading student work.

> "Write a BASIC program that reads in three data items and prints them in ascending order".

The instructor must first build a student name file. As an illustration, consider the creation of the following student name file:

```
     / LØGØN Δ........
     / EXEC Δ EDT
     1.0000  NAME1
     2.0000  NAME2
     3.0000  NAME3
     4.0000  NAME4
     5.0000  @ SAVE 'any name' : 1-129
```

Next a data/result file is generated by the instructor. This may be created in EDT as follows:

```
     1.0000  12 (defines the # of input
                 data terms)
     2.0000  1,2,3,-1,-2,-3
     3.0000  2,0,-2,1,-3,1
     4.0000  12 (defines the # of output
             data items)
     5.0000  1,2,3,-3,-2,-1,-2,0,2,-3,1,
             1
     6.0000  @ WRITE 'DATA1'
```

With this information the GRADER is equipped to grade student programs. Specifically suppose student, NAME4, composes the following program for printing three data items read in ascending order.

```
     *10  READ A,B,C
     *20  IF B <= C THEN 60
     *30  LET T = B
     *40  LET B = C
     *50  LET C = T
     *60  IF A <= B THEN 100
     *70  LET T = A
     *80  LET A = B
     *90  LET B = T
     *100 PRINT A,B,C
     *110 GØ TØ 10
     *900 DATA 1,2,3
     *901 DATA 2,4,1
     *999 END
```

In order to use the GRADER the user must save the program as follows:

```
     * SAVE DRCØNE
                  It is good practice
                  to start the name
                  with the three
                  initials of the
                  student's name.
     *BYE
```

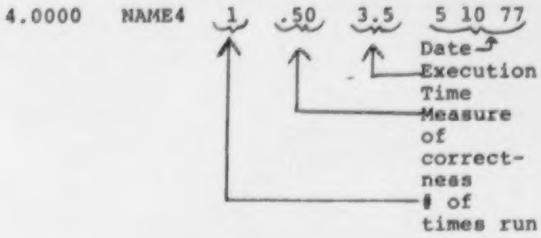
In the executive command mode type the following command:

```
     / DØ GRADER, (DRCØNE, 1,4,NAME4)
```

The GRADER would execute program DRCØNE. The READ statement would automatically access data from the data/result file NAME1. The system would generate the following output for student use:

```
      1        2        3
     -3       -1       -2
     -2        2        0
     -3        1        1

      Out of Data
      2 out of four were correct
```

At the same time the GRADER would record this information in the student name file as follows



3. The impact of the grader upon the teaching and learning of programming

The purpose of this section is to report on our experience of using this automatic grading system in a large university environment at Georgia State University. Our discussion will reveal the power of such a tool in the teaching and learning of programming and hopefully provide some insight about the organization and structure that is necessary for an effective use and application of such a system.

It is a fact of life that most people are reluctant to accept change. This attribute of human beings makes it difficult to introduce a new system. We also know that a necessary attribute for a system to be eventually accepted and used is that it must be easy to use. Our experience has shown that simplicity and convenience should be a design objective which can be met by structuring the system into a few basic functions that are relatively independent. This is the basis of the three clear subdivisions of our system as shown in figure (1). From the students point of view the system is not only simple and straightforward to use but takes the burden of supplying data. This feature can be exploited for teaching the science of testing to validate the correctness of programs.

The system from the instructor's point of view is also relatively straightforward and easy to use. However it places an extra burden on the instructor to create comprehensive data files to fully test student programs. The task is demanding and it often discourages the instructor to make use of this system. We decided to provide this service to the instructors at Georgia Georgia State University. This led to the development of a program library containing most of the favorite problems of the instructors together with well designed test data files. This facility besides providing the instructor a repository of good problems for student assignments, has initiated a trend towards better coordination and standardization of the material covered by different instructors in the various sections of IS 301. Since the system has been in operation for less than a year, it is too early to assess its full impact upon the teaching of programming. On the other hand, although we have not conducted a scientific survey of the students to assess its impact upon the learning of programming, yet informal student comments have convinced us that the system is regarded as a useful addition to the learning aids.

References

1. Chand, Donald R. "Automatic Grading of Computer Program" Georgia State University, May 1977.

2. Chand, Donald R. "Automatic Grader Users Guide" Georgia State University, May 1977.

3. Forsythe, G. E. and N. Wirth "Automatic grading programs" Comm. ACM 8, 5, May 1965. (pp 275-278)

4. Temperly, J. F. and B. W. Smith "A grading procedure for PL/I student exercises" Computer Journal 10, Feb. 1968 (pp. 368-370).

COSTS AND DESIGN CRITERIA FOR CATC SYSTEMS

Stanley Etersque
Division Of
Systems and Computer Services
Medical College of Georgia

Abstract

This paper describes results related to the design criteria and costs associated with two CATC systems, one batch-oriented and the other interactive. With both systems utilizing the same items, cost data was recorded at every step in the processes that related to storing the items, writing the programs, generating the tests, and recording the test results. Based on the results of the project, criteria are presented for the content of item banks and for several factors related to interactive testing systems.

INTRODUCTION

The growth rate of CATC systems has been such that in 1973 there were at least 22 in existence, the first reported one having been well established by 1966; at this conference 12 new ones are being described, and we could probably not easily identify all the others in use throughout the U.S. today.

The nature of these CATC systems varies greatly across a broad spectrum of utilization philosophy, subject matter area, delivery system, and maintenance features. This paper will focus on issues related to the development of CATC systems that will satisfy the needs for convenience and beneficience to both students and faculty. To this end, design criteria will be presented that should be considered by those who will be developing new CATC systems, so that more orderly progress can be made in satisfying the needs for which such systems have been implemented.

The primary basis from which these criteria have been formulated was a project the author completed which compared the benefits and costs of two CATC systems. One system, the Test Administration and Scoring System (TASSY), which was developed by T.L. Blaskovics and J.A. Kutsch at West Virginia University, had been in use for two semesters and tested students interactively; the other system was designed by the author for this project, had been used for one semester, and produced tests that were administered in the traditional classroom, or "batch" mode. The objectives of the project were to: (a) determine the costs for developing and using such systems; (b) determine if self-scheduled interactive testing would significantly improve learning; (c) determine if repeating tests after an externally prescribed period of time had elapsed would significantly improve learning; (d) determine if both taking tests and repeating them interactively would significantly improve learning as compared to taking and repeating tests in the batch mode, and with the same external prescription for elapsed time before repetition. Objectives (b) through (d) were measured by student performance on a comprehensive final examination in the course, which was introductory statistics.

PREPARATION FOR THE PROJECT

An item identification scheme was set up that would facilitate the use of the same set of items in both the batch and interactive CATC systems with a minimum of conversion effort. Besides the attributes of each item, the item identifiers contain the correct answer.

The author gathered and selected test items from statistics textbooks and previous tests given by those who had taught the introductory statistics course for the Department of Statistics and Computer Science at West Virginia University. During the semester prior to the execution of the project, all items were classified by chapter and subsection number by the author to match the textbook used in the course. They were also evaluated as to their clarity, possible ambiguity, and appropriateness. Items thus selected were interactively entered into a computer disk file and were edited interactively as needed based on several proof-readings of them. The interactive terminal language used, WYLBUR, allows upper and lower case characters and has excellent text editing capabilities. Both of these features are very desirable for CATC systems and eliminated many other problems from arising.

This pool of 616 items was used to generate 15 tests with the batch CATC system, for five instructors of the introductory course during that Fall semester prior to the actual project. The author generated the tests and scored them by using the punched card output from an optical scanner with two different scoring programs. One scoring program was within a service provided by the Computer Center and the other was written by the author to additionally perform psychometric analyses on the items which then served as the basis for keeping, rejecting, or modifying the items to be used in the Spring semester study.

The instructors submitted requests for tests based on the number of items they desired for each subsection of each chapter from the textbook in use for the course. A table of random digits was used to randomly select items from the pool that would satisfy the request. Initially, the same item could appear on two different instructor's tests over the same subsection material. This was modified after two tests had been generated, however, so that as many different items as possible could be selected and submitted to analysis, since no statistics about the items were available. The items selected were submitted to a series of programs that reformatted them and stored them on a disk file. They were then retrieved at a terminal under control of the WYLBUR language so that ditto masters could be used for the subsequent mass production of the tests.

Items which were then carried over in the pool for the Spring study had to achieve a point biserial value of 0.30 or better. Other items carried over included new ones that were entered for "sparse" subsections during the Spring semester. In addition, all items carried over were screened and modified, where necessary, to contain 500 or fewer characters. This limitation was imposed by the nature of the interactive system but only necessitated modification of three items. Ultimately, 238 items were used in the interactive CATC system.

The times required for the various activities related to the two CATC systems were recorded throughout the study. One time factor recorded was CPU time, and the other was elapsed, or real time. These two temporal classifications were recorded for terminal entry and editing of items, test generation, test and item analysis, restoring and updating gradebook files of interactively tested students, and system security functions necessary for the interactive system. A fundamental assumption was made that the time involved in item writing/selecting and categorizing is going to be constant for the individual in a given discipline regardless of the testing system, computerized or manual, that is employed. Therefore, such times, though they constitute a very real indeterminable factor in test construction, were not recorded for this project.

CARRYING OUT THE STUDY

The study was conducted in three sections of an introductory statistics course during the Spring 1973 semester. Two sections were taught by a graduate assistant and the third by the author. A questionnaire was administered to the three classes at their first meeting to capture data that might relate to the effects on learning of the two CATC systems. This data came from questions about the students' backgrounds in mathematics, statistics, and computer science in high school and college. In addition, ten arithmetic and ten algebra items from a previous study of factors affecting success in statistics (1) were administered with the questionnaire. Additional data was obtained from the Registrar on class rank and major. Sixteen experimental combinations were formed based on four variables: sex, upper or lower division standing, science or non-science major, and statistics instructor. The students were then randomly assigned to one of the four groups, three experimental and one control, established for the study. These groups were: 01-interactive testing without the option of repeating tests; 02-batch testing without the repeat option (control group); 03-interactive testing with the option of repeating each test once; 04-batch testing with the option of repeating each test once.

Students who could not or did not repeat tests were given that one score on their tests; those that repeated had the two scores averaged together for theirs. All three classes were taught using conventional lecture-discussion methods, used the same textbook, and omitted identical sections of material. Students in the interactive testing groups did not come to class on the days that batch tests were administered, though they individually scheduled their own interactive testing times so as to complete all testing within a two week period over each given unit of material. Repeat batch tests were given during the second week of each period and repeating interactive students had to wait at least three days before taking their repeat test. All tests for all students were composed of the same number of items within each unit. The initial batch tests for the three classes all had exactly the same items, though they appeared in different sequences for each class. No item was permitted to reappear on the repeated batch test, though there was no such stricture possible with the repeated interactive tests. In addition, since each interactive test item was randomly generated based on its attributes, there was no uniformity among the interactive tests for any unit. Students taking interactive tests were required to sign up in advance so that terminal time for them would be guaranteed.

Numerous computer breakdowns during the study necessitated extensive recovery efforts for completing interactive tests, updating gradebook files, and recording times involved for all aspects of building and using both CATC systems. The time for these activities, as well as for those relating to administering tests, were recorded as the study proceeded.

RESULTS OF THE STUDY

The null hypotheses related to achievement differences between batch and interactive groups and repeat/non-repeat groups that were formulated could not be rejected at the 0.05 level.

The aborted interactive test rate in the study was 30%. That is, 30% of the 219 interactive tests that were attempted had to either be completed by batch processed methods or by taking subsets of the test units interactively at a later time.

It was discovered that instructional philosophy and orientation of the questions in the item pool have an impact on the achievement of students being tested with items from the pool. That is, the graduate assistant emphasized computational skills but the test items were of a conceptual nature. Thus, students in one of her classes who consistently prepared for computational problems to appear on their tests, based on indications of strong rapport between that class and the graduate assistant, failed to achieve as well as the other two classes. The result, then, was that these students were misled in their test preparations, much as other students would be in a similar situation where an instructor utilized items from a bank of questions that were prepared by people with an orientation toward questions entirely of a conceptual or a computational nature while the instructor was inclined toward the other extreme.

The costs for the two CATC systems utilized in this study are tied to the various activities involved. The activities associated with the batch system were:

1. Item entry into the pool.
2. Item modification.
3. Item sorting based on subject matter categories.
4. Item selection for the defined test.
5. Printing of the test and key for the instructor.
6. Reformatting items prior to final printing.
7. Printing the test on ditto masters at an interactive terminal.
8. Storing item identifiers for use in item analysis.
9. Production of tests from the ditto master for student use.
10. Utilization of instructor and class time to administer tests.
11. Test scoring, item and test analyses.

Activities 4 through 11 were performed for each unit test that was produced. Activities 1 through 3 were also performed in using the interactive system. The remaining activities necessary in using the interactive system were:

1. Defining tests and subtests.
2. Scheduling, proctoring, and assisting students to take interactive tests.
3. Retrieving the file that indicated attempts to take interactive tests.
4. Retrieving the student grade book file.
5. Retrieving the file of actual item numbers, correct responses, and student responses to the generated test items.
6. Restoring and completing the records of students who were unable to complete tests before a system malfunction terminated their sessions.

Table I summarizes the costs associated with each of the CATC systems. In the categories where actual association with one system or the other is impossible to determine, the costs have been estimated on a pro-rata basis. The computer utilized was an IBM S/360 model 75 with 512k bytes of high speed 4-way interleaved core memory, 1024k bytes of low speed core storage and numerous peripheral devices.

There are no costs shown in Table I for Test Scoring with the interactive system because that activity is handled as a routine part of the system during each testing session. The Divided Cost/Test figures were arrived at by dividing the number of student tests administered into the Total Cost when Item Maintenance was equally divided between the two systems. The Cost per Test was arrived at by using the Total Cost that would result from charging all Item Maintenance to each system.

Note that the Costs per Item, $ .44 and $ .70, are fairly high. The Divided Costs/Item, $ .27 and $ .53, compare favorably though, with those that can be expected from Salisnjak's (2) figures when the rates used in arriving at Table I are applied to the manual production of a 15 page, 75 item multiple choice test. The figure then would be $ .36 per item. In addition, with repeated use of the CATC systems, the cost per item would continue to drop while the manual production costs would stay the same each time that method is used. It should also be noted that, aside from item maintenance costs, the most costly items associated with the interactive system are personnel for test administration and computer time for record-keeping. Both of these costs can be reduced, however. Interactive test administration costs can be reduced by utilizing a larger number of terminals in one centralized area - an economy of scale reduction. Record-keeping costs were excessively high because of the high rate of aborted test sessions caused by frequent host-computer system crashes.

SUGGESTED CRITERIA

The criteria proposed for incorporation in future CATC systems are based on the author's experiences with two such systems, one batch-oriented and the other interactive. The author believes that only through a continual process of incorporating design factors that have been observed to have been needed in previous CATC systems can the state of the art advance without the waste that comes with everyone "re-inventing the wheel" when designing such systems. The criteria are presented below in three major areas related to CATC systems: 1) CATC system support; 2) test related features; and 3) the item pool. It will be noted that some relate only to interactive systems, while others, when not so specified, relate to both batch and interactive CATC systems.

A. CATC System Support

Each of the systems used in the study have some very fine features which make their utilization quite easy for an instructor. The minimal features for CATC systems are:

1. acceptance of upper and lower case letters (for better conformity to class-and text-presented material).
2. the ability to modify only the parts of items in need of it, instead of having to re-type the entire item.
3. the ability to define tests (with interactive systems) or select items (with batch ~~systems~~) by both batch and interactive methods, depending on the instructor's desires as he uses the system.
4. the capability to get analyses of the generated interactive tests and items. This will require the development of new techniques and standards for non-uniform tests.
5. the ability to retrieve the generated interactive test, on demand, in an abbreviated form that will facilitate discussing student challenges to questions.
6. the ability to retrieve a file of attempts to take interactive tests, in order to spot student difficulties in using the system. Also, a related file should be built from this one that will keep statistics on system usage and failures.
7. before any interactive system is utilized routinely, it should be subjected to simulated runs on the host computer so that expected completion rates can be determined. This would result in far less frustration during actual use if a sustained completion rate of 90% could be reached during several points in the academic term and in spite of the host-computer configuration modifications that occur so frequently.

B. Test Related Features

These are the features that are deemed to be necessary during or immediately following test-taking:

1. the ability for the student to skip an item until later in the interactive testing session.
2. the ability for the student to have an item reprinted on demand. This is desirable when paper jams or line-noise distorts transmission.

TABLE I

Cost Comparisons for Batch and Interactive Systems[a]

Cost Item	Batch System[b]	Interactive System[c]
Item Maintenance		
Personnel	$ 451.37	$ 451.37
Computer	213.05	213.05
	$ 664.42	$ 664.42
Test Generation		
Personnel	$ 107.64	$ 9.70
Computer	41.88	1.96
Supplies	4.65	---
	$ 154.17	$ 11.66
Test Administration		
Personnel	$ 50.28	$ 873.54
Computer	---	156.11
	$ 50.28	$ 1029.65
Test Scoring		
Personnel	$ 130.14	---
Computer	84.12	---
Supplies	2.54	---
	$ 216.80	---
Record-Keeping		
Personnel	$ 4.00	$ 13.47
Computer	---	363.86
	$ 4.00	$ 377.33
Total Cost[d]	$ 1089.67	$ 2083.06
Divided Cost/Test	$ 5.00	$ 9.69
Cost per Test	8.05	12.78
Divided Cost/Item	.27	.53
Cost per Item	.44	.70

[a] Figures shown are for three unit tests. Batch groups had three versions for each and three makeup tests.

[b] 218 student tests administered; 3,980 items.

[c] 215 student tests administered; 3,920 items.

[d] If item maintenance costs were charged to each system instead of dividing it equally as shown above, the respective costs would be $1754.09 and $2747.48.

3. the ability to have responses verified by the system on the student's demand. That is, for example, when in answering a true/false item with a T instead of a 1 (for true), the student should be informed that his response is not appropriate.
4. the option of having the correctness of each item response appear immediately upon answering or at the end of the entire interactive test. This feature could be of great value in controlling student anxiety.
5. automatic recording of item statistics in a file at the time the item is scored, whether the test is taken interactively or in batch mode.
6. automatic gradebook recording of student scores when the test is scored, whether taken interactively or in batch mode.
7. immediate recording of the question number, correct response, and student answer as soon as the student has responded to each item. This would enhance the availability of feature 5 in A, above, even in the event of aborted tests due to system crashes.

C. The Item Pool

1. A study should be undertaken to accurately define the role of the item pool, other than just its size. Such a study should be designed to determine the effects of testing for one aspect of instruction but teaching for another, such as the distinction between problem-solving skills and the understanding of concepts. An interesting variable to consider is the effect on students who have a strong rapport with the instructor in such a situation. The issue of item pool content appears to be one that educators will have to grapple with more and more as item pools are exchanged or are developed by several people. Perhaps the individual instructor needs to make his own decisions on the type of pool to use according to his own teaching style, regardless of what subject matter specialists or learning theorists might advocate. It seems that inevitably the instructor will be presenting material based on his own learning strategy or his relative knowledge of the subject matter; i.e., if he is well-grounded or found the material easy and non-threatening when he was learning, he might be inclined to teach for understanding of concepts, otherwise for problem-solving skills. This issue has implications for standards of item classification which might best be achieved by having independent judges classify items according to their content.
2. Early in the design of an interactive CATC system, the question of whether or not to provide diagnostic feedback should be resolved. If it is decided to do so, it would involve keying each item in the pool to particular page references for the material in a textbook or to cross-reference a list of texts that include the same material as the item. This approach is possible by generating items in pairs, the second part being the diagnostic information for the item with prescriptions for each incorrect distractor. Another possibility is to store the diagnostic information with the item itself or with the item identifier. Some serious drawbacks to the issue of providing diagnostic feedback must also be considered. Providing such information requires an enormous amount of additional effort at the same time as the items are prepared for inclusion in the item pool. In addition, to key items to a single text virtually forces instructors to either not share their pools outside their own local institutions, or requires that anyone else wishing to use the pool use the same text. Furthermore, changes of texts, either by new adoptions or by virtue of subsequent editions, obsoletes the prescriptive information, forcing a need to have such feedback suppressed, ignored, or revised at a great cost. It would appear that the best solution to providing diagnostic feedback, if it is so desired, would be to include cross-referenced indices within the item pool that are keyed to distributed lists of the current textbook(s) available to students.

REFERENCES

(1) Hale, R.E. An analysis of variables related to achievement of graduate education students in introductory inferential statistics. Unpublished doctoral dissertation, West Virginia

University, 1970.

(2) Salisnjak, J. Computer aided test preparation: Six years of experience. *Educational Technology*, 1973, *13* (3), 37-38.

ITEM SELECTION: A COMPUTERIZED DECISION-MAKING PROCESS

Gregory C. Hamilton
Office of Minority Student Education
College of Engineering
Michigan State University

A test construction system is described which employs a computerized decision-making process for item selection. Test items are classified by content and structure variables allowing the user explicit control in test construction. The system is designed for small enrollment, mastery model mathematics courses. Item construction incorporates random number generation in conjunction with strictly controlled algorithms. Design and operational performance of the program is discussed along with item construction and cost analysis.

Introduction

During the past decade the interest in and usage of computer-assisted test construction (CATCON) systems has grown rapidly. As computer technology increases so will its application in the area of education. In 1974, the College of Engineering at Michigan State University, funded by a grant from the Alfred P. Sloan Foundation, undertook the task of developing a series of eight courses designed to supplement the educational backgrounds of minority students. The courses were written in the self-paced mastery model and each used a computerized testing system. Several existing programs were reviewed and the faculty decided to use a fixed-format system originally developed by Dr. Franklin Prosser at Indiana University (1970). This program provided for exams which were: 1) easily produced; 2) errorless; 3) individualized (unique); 4) secure from unauthorized access; 5) easily modified; and 6) economical.

The self-paced instructional material is divided into a series of modules. Each module is an aggregate of several objectives, and for each objective 10 - 15 test items were written. A group of test items pertaining to a single objective constitutes a test bank. The group of test banks which compose a module exam is called an item pool or item bank.

At the conclusion of the three year grant period, ending in July 1977, the entire testing component consisted of 7,523 test items. Despite its size, the testing component did not contain the flexibility needed to meet the overall goals of the project. Three major problems were encountered:

1) Maintaining this large number of test questions was impractical for the small numbers of students involved. The process of writing, coding, proofreading, and modifying the questions was extremely time consuming and expensive. And since there were roughly 20 students per term per class and each student answered approximately 100 exam items, the frequency distribution for any particular item was very small (in some cases zero). This particular testing system is more appropriate for large enrollment classes where optimum use of each item is possible.

2) The items within any test bank tended to be isomorphic. Differences between items were restricted to numerical variations or modification of key textual phrases. Within the mastery model where repeat testing is allowed, this isomorphism led to a "study the test" behavior on the part of the students. The testing system produced similar exams, thus encouraging students requiring more than one test to concentrate on question forms and not necessarily on the underlying concept illustrated by the item solution.

3) The rationale for writing isomorphic test items was to eliminate the need for external controls over the selection process. An underlying assumption for item selection was that even though there may be minor differences in item difficulty within a test bank, these differences would statistically average out through random selection. In fact, supposedly equivalent tests proved to be radically different in their respective difficulty. Student performance was measured against a preset criterion level of 80%. However, since the exams were not consistent in their degree of difficulty, there were many cases where a student reached the 80% criterion level becuase the exam contained

relatively easy items.

At the commencement of the Fall 1977 term, the Department of Mathematics and the Office of Minority Student Education within the College of Engineering provided support for continuing two of the courses developed under the Sloan Foundation grant. These courses are College Algebra and College Trigonometry. After a critical evaluation of the testing component of these courses, it was decided to rewrite the computer program and the item banks. The new CATCON system was designed around the following seven objectives:

1) to provide maximum flexibility in test content and structure;
2) to provide maximum flexibility in item content and structure;
3) to accommodate strict external control on item and test structure;
4) to incorporate a simplified method of item and test modification;
5) to reduce the number of test items to be prepared;
6) to produce tests at costs comparable to the fixed-format systems;
7) to include all computer functions available in the fixed-format systems.

Selection Process

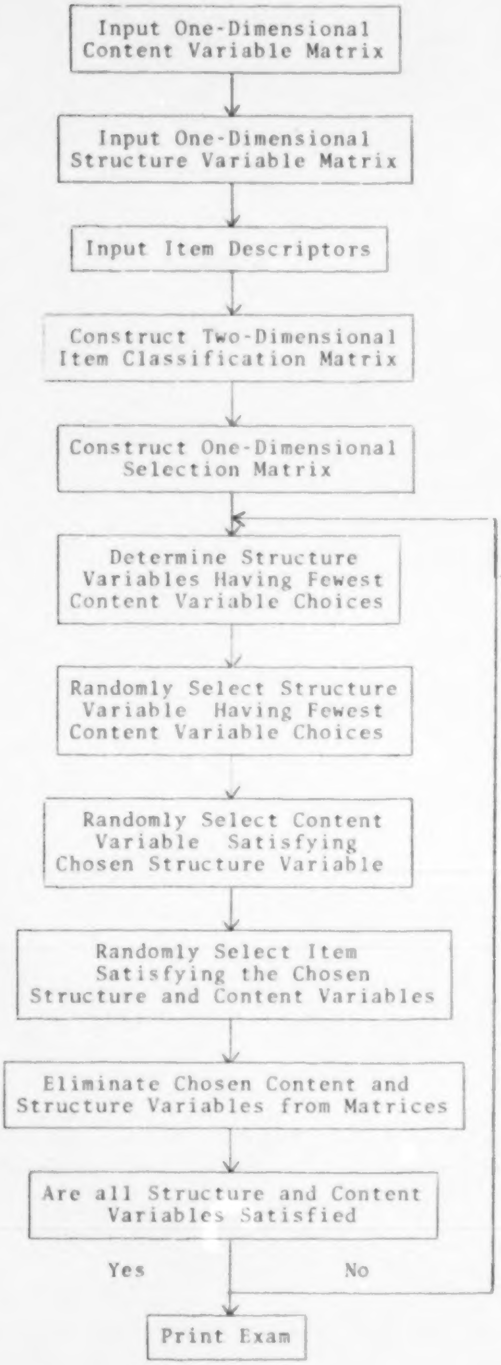
The main focus of the program was to provide the user with a systematic method of explicitly specifying the test content and structure. Test items are classified along two dimensions, which are set by the user.

Several other systems employ item classification. The CGRT program developed by Prosser (1973) classifies items according to topic or concept. This system was used for the Sloan Project and items were randomly selected from the test banks. The MENTREX system (Libaw, 1973) sets up a two dimensional matrix and randomly selects questions according to the two variables specified by the matrix. The MENTREX system, however, is designed for multiple-choice or essay questions and produces tests which use mark-sense forms for computerized scoring. Two other widely used CATCON systems are the CTSS system (Toggenburger, 1973) and the MEDSIRCH system (Hazlett, 1973). Both of these systems utilize item classification along several dimensions. (The MEDSIRCH program allows each item to be classified according to a maximum of 57 different descriptors.) These two systems are designed for multiple choice tests, and contain several thousand items. The two-step selection process involves a scanning procedure which first creates a small item sub-pool. This sub-pool contains test items which satisfy the prescribed attributes desired by the user. The test items are then randomly selected from this sub-pool. There are several other systems that have been developed and Stodola presents a brief description of these programs in a collection of monographs published by Lippy (1974). These systems require a very large number of test items and are not particularly appropriate for use in small enrollment classes.

The CATCON system developed at MSU utilizes item descriptors to construct a two-dimensional matrix from which test items are selected according to a prescribed algorithm specified by the user. The user controls the item descriptors and the selection algorithm through the program input. Figure 1 illustrates the logic diagram of the program. The user inputs two composition codes. The first indicates the test content variables. These content variables are cross-referenced in a one-to-one correspondence with the instructional objectives. The second contains the test structure variables indicating the number and type of items to appear on the test. These variables can assume any representation desired. They may indicate the item form (true-false, multiple choice, short answer, etc.), item difficulty index, item point value, cognitive level or any other type of item attribute which may be used for item selection. Following these composition codes, the item descriptors are input. Arranged by content variable, the item descriptors themselves represent the structure variable. A square two-dimensional item classification matrix is formed containing the number of items satisying both content and structure variables. Figure 2 represents an example of these matrices, and illustrates that the tests are to contain nine questions, one from each of the content variables indicated. Furthermore, the test must contain one item having structure variables 1, 2, 3, and 10; two items with structure variable 5; and three items with structure variable 15. The resulting selection matrix contains the number of content variables which will satisfy the structure variable. Structure variables 1, 2, and 3 can only be satisfied by one of the content variables, while structure variables 5, 10, and 15 can be satisfied by 4, 6, and 6 content variables respectively.

The actual selection of items is accomplished by an iterative process. The selection matrix is scanned to find the structure variable with the least number of content variable choices available. (Should there be more than one with the same number of choices, the program randomly selects one to start with.) In the example, three structure variables have only one possible content variable choice. For structure variables 1, 2, and 3, the decision is automatically made: structure variable 1 is satisfied

Figure 1 - Logic Diagram


Input One-Dimensional Content Variable Matrix
Input One-Dimensional Structure Variable Matrix
Input Item Descriptors
Construct Two-Dimensional Item Classification Matrix
Construct One-Dimensional Selection Matrix
Determine Structure Variables Having Fewest Content Variable Choices
Randomly Select Structure Variable Having Fewest Content Variable Choices
Randomly Select Content Variable Satisfying Chosen Structure Variable
Randomly Select Item Satisfying the Chosen Structure and Content Variables
Eliminate Chosen Content and Structure Variables from Matrices
Are all Structure and Content Variables Satisfied
Yes
No
Print Exam

only by content variable 4, 2 by 8, and 3 by 18. The particular question for the structure-content combination (1,4) is a random choice of one of the six available questions. Once the three questions for structure-content combinations (1,4), (2,8), and (3,18) have been chosen, the selection proceeds to structure variable 5. The program will randomly select one of the structure variables with 4 content variable choices , randomly select one of the four content variables available, and finally randomly select one of the items from the allowed number given in the item classification matrix. The process continues until all structure-content combinations have been determined.

After each test item has been chosen, all matrices are reduced appropriately. The item classification matrix has the chosen structure variable row and the chosen content variable column eliminated. The matrix, therefore, remains square, and contains information only related to those items yet to be determined.

Once all items have been selected, the exam is printed. The user has the option of printing the same exam as many times as desired or recycling and beginning the process over again. This second option produces unique exams and is particularly advantageous for the mastery model instructional format where repeat testing is available.

The selection process does not guarantee that the same structure-content variable combinations will appear for every test produced. The decision-making process, however, does guarantee that tests produced are equivalent with respect to their overall content and structure.

Test Flexibilities

The content and structure composition codes and the item descriptors are entered through the input data. Therefore, the user can modify either the content or the structure of the tests by simply changing the input values. The program itself is insensitive to the numerical values assigned.

Test content is easily modified and tests containing more than one question per concept can be produced by specifying the particular concept variable as many times as desired. A particular concept can be eliminated from the test by eliminating the concept variable from the composition code. Likewise, the structure of the test is modified by appropriate adjustments in the structure composition code.

Often, particular items are judged to be inappropriate for selection. To prevent the item from being chosen, the user assigns an item descriptor which is not one of the values specified by the

Figure 2 - Illustration of Matrix Construction

Content Variable Matrix	3	4	6	8	11	13	15	18	20	Structure Variable Matrix	Selection Matrix
Item Classification Matrix	-	6	-	-	--	--	--	--	--	1	1
	-	-	-	4	--	--	--	--	--	2	1
	-	-	-	-	--	--	--	3	--	3	1
	-	-	4	-	--	1	1	--	1	5	4
	-	-	4	-	--	1	1	--	1	5	4
	3	-	1	-	2	4	2	--	1	10	6
	2	-	2	-	4	1	1	--	1	15	6
	2	-	2	-	4	1	1	--	1	15	6
	2	-	2	-	4	1	1	--	1	15	6

selection composition code. This ensures that the unwanted item will never enter into the item classification matrix, and therefore will never be selected. This technique of deleting items foregoes the process of item bank modification required by several other programs.

The program structure provides an interesting blend of control and flexibility over test construction. Use of the composition codes and 2-way classification of items allows for considerable flexibility in the content and structure of tests; the decision-making process employed provides the necessary control over test construction to guarantee balanced exams.

Item Construction

The item classification matrix of Figure 2 is used to construct one of the tests for the College Algebra course. The item bank for this test consists of 44 questions. This is a relatively small number compared to the 125 items for the same test in the fixed-format system previously used. This reduction is made possible through the use of random numbers generated under the control of strict algorithms. The random generation of numerical values and/or key textual phrases within each item reduces the number of items that need to be prepared, but does not affect the quality of the exams. The reduction in the number of items permits the development of a greater variety of item structures.

Each item consists of four basic operations: 1) defining allowed ranges for the random variables; 2) random generation of variables; 3) defining all output variables by the item algorithm; and 4) item printing via a FORMAT statement. The item itself in actual practice is basically a small program. It may be of the fixed-format type, where only WRITE and FORMAT statements are necessary. Or, items may be very complicated and include a large number of programming statements and complex algorithms. For example, one question concerning expressions dealing with logorithms is a three part item. Each part has six possible forms from which one is randomly selected. Therefore, a total of 6! or 720 combinations are possible. The fixed-format system previously used contained only 15 such items. It should be noted, however, that this increased flexibility in item construction provided through the use of the random number generator is accompanied by increased complexity in the item programming.

The items are printed according to WRITE and FORMAT statements. Problem statements are printed in the first 85 columns and answers appear in columns 86 through 135. With this spacing arrangement, exams and answers are separated at the 8½ inch mark.

This type of item programming can be designed to generate random variables and work toward the solution, or generate the answer and work backwards. Variable formatting options allow for complex problem statements to be constructed easily. In general, the random generation of problem components reduces the total number of items while increasing the number of possible equivalent problem statements.

Development and Analysis

Presently, ten exams have been written using this CATCON system. The exams are written in the FORTRAN IV language for use on a CDC 6500 computer. The physical size of the programs ranges from 600 to 900 cards. Each exam required approximately two weeks to write, debug, and begin test production.

Development costs vary depending on the programmer's experience and the number of items that are prepared. For the ten module exams prepared, this averaged between $15 and $20. The cost per exam depends on the number of items constructed and the complexity of the item algorithms. Cost per exam for this system is between 4¢ and 5¢. This is slightly higher than the fixed-format exams, which are approximately 3¢ to 4¢ per test, but is still reasonable compared to clerical time and/or standard reproduction costs.

Statistical analysis procedures are an integral component of the testing system. Statistical analysis programs have been written which provide valuable information on how the items are received by the students, and the effectiveness of the repeat testing option in the mastery model. Item means and standard deviations are computed, providing insight into the difficulty of particular items. Through the use of statistical analysis techniques, the item descriptors are being refined to provide another dimension along which items may be classified and selected.

Conclusion and Summary

The program developed at Michigan State University has provided the flexibility and versatility which were the bases for its design. Despite resolving the three major problems encountered with the fixed-format CATCON system, this program has three major disadvantages of its own. The item classification matrix shown in Figure 2 contains a number of empty cells. These cells can be filled by the creation of items which possess the structure-content combinations required. Using a matrix with cell vacancies, though, does not present a problem. The user must be careful in assigning item descriptors in order to avoid the possibility of an item classification matrix which may not provide for all of the structure-content combinations requested. The program is insensitive to the numerical values assigned to the items and to the composition codes entered by the user. This indifference permits a redefinition of variables and provides the user with a simple but effective method of controlling the selection of test items.

The second disadvantage rests with the fact that the program is a FORTRAN routine and must be compiled. The size of the program and the number of items will eventually be limited by the amount of core available for compilation. As the number of items increases, it will become necessary to create individual subroutines for each one of the content variables. However, such is the long range goal of this program. Each concept for an entire course or series of courses would be represneted by a single computerized test bank of questions. These small test banks would be attached to the main selection routine in any order, and across instructional units or even course boundaries. This type of option is most appropriate in mathematics where continual review is often necessary. At the present time, the exams are constructed within a narrowly defined set of instructional objectives and the item banks have not reached the core space limit.

The third disadvantage with the program centers on item modification. Any change in the text of an item usually involves the rewriting of a FORMAT statement. Changing the item algorithm involves determining which output variables must be changed and the easiest way to make the change. It is not uncommon for even the slightest modification to require a complete reprogramming of the item. Eliminating an item from the item classification matrix is easy, but structural modification of an item requires some, if not a complete, reprogramming effort. Consequently the program must also be recompiled. This problem can be avoided in most cases by carefully constructing the item prior to programming.

These three disadvantages are restricted to the computer programming aspects of the system, and are not associated with the quality of the exams. However, some concern has arisen over the structure-content combinations of the items. These combinations are not fixed across all exams produced. However, our foremost concern is that the exams be consistent relative to the criterion level. The variability of the structure-content combinations across exams would pose a serious problem if a criterion-level was established for each concept. Presently, the mixing and matching of the structure-content variables provides for exam flexibilities which are enthusiastically accepted by the course instructors and lead to a shift in student attitudes with respect to testing.

References

Hazlett, C.B. "MEDSIRCH: Multiple Choice Test Items", Educational Technology, 1973, 13(3),24-26.

Libaw, F.B. "Constructing Tests with the MENTREX Tutorial Testing System", Educational Technology, 1973, 13(3),30-31.

Lippey, G. Computer-Assisted Test Construction, Educational Technology Publications, Inc., Englewood Cliffs, New Jersey, 1974.

Prosser, F. Research Computing Center, Indiana University, Fall, 1968. Major Revision July 1970. Copyright, 1970.

Prosser, F. "Repeatable Tests", Educational Technology, 1973, 13(3),34-35.

Toggenburger, F. "Classroom Teacher Support System", Educational Technology, 1973, 13(3),42-43.

TEST GENERATOR

Lois J. Keller, R.S.M.
Robert L. Williams
St. John's University
New York

In the Education Resource Center of the College of Pharmacy and Allied Health Professions at St. John's University four programs are used by the faculty to assist in the preparation of tests and quizzes. The programs, and the files necessary for their operation, are simple, allowing instructors to devise and prepare privately and without aid, spirit duplicator masters for test and quiz production.

Preparing quizzes, tests and examinations for large class sections or multiple class sections often proves to be a difficult clerical task rather than a professional one. Even when a large quantity of well-constructed test items have been sufficiently prepared in advance the clerical tasks of typing and producing the examination copies are tedious. This is especially true when it seems advisable to administer a number of forms of a test to one large section or to each of several sections of a course. And where make-up quizzes are anticipated because of student absences, the tasks become even more difficult. In addition, where technical terms that compose the content items are not familiar to the typist, the chances of misspelled or misrepresented items are usually increased and the time required for typing is lengthened. In effect, even with the best bank of sound test items, the instructor often finds that the difficulties of test production preclude the efficient and professional preparation of tests and quizzes.

Described below are four programs which may be of significant help to those who have on-line computer terminal capabilities available to them and who face test preparation problems similar to those noted. In the College of Pharmacy and Allied Health Professions at St. John's University these four programs are used to prepare spirit duplicator masters for quizzes and tests on Decwriter II terminals. These terminals accept perforated spirit masters in a manner similar to regular terminal paper. When these programs are run, they allow the instructor to title and code each form or version of his quiz. Where a quiz is longer than one page, each is paginated. As soon as the master is removed from the terminal it may be carried to a duplicator for immediate production. No typist need be involved.

All four programs address 'test' files prepared by the instructor (or his assistant). From one test file--for example a file developed for a six-week unit of studies--a variety of multiple-choice and true-false tests are randomized. Two short example runs of TESTA, which produces multiple-choice tests and answer keys, follow:

```
TESTA

TYPE THE NAME OF YOUR TEST FILE.  ? 001
001 HAS 50 ITEMS, PROFESSOR.
PROFESSOR, COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE A
PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS!  ? 3
WHAT FILE QUESTION SHOULD I BEGIN WITH!  ? 11
WHAT PATTERN SHOULD I FOLLOW!  1 EVERY QUESTION
2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ? 2
PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER.
ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.?

                    EXAMPLE A                              1

  1 . THE RELAXATION PHASE OF THE CARDIAC CYCLE IS CALLED:
        1 .  DIASTOLE
        2 .  ARTERIOLE
        3 .  SYSTOLE
        4 .  REFRACTORY

  2 . THE DEVICE FOR MEASURING BLOOD PRESSURE IS:
        1 .  STETHOSCOPE
        2 .  CARDIOMETER
        3 .  SPHYGMOMANOMETER
        4 .  ELECTROCARDIOGRAPH

  3 . THE BLOOD ENTERS THE RIGHT ATRIUM THROUGH:
        1 .  SUPERIOR VENA CAVA
        2 .  ASCENDING AORTA
        3 .  DESCENDING AORTA
        4 .  INFERIOR VENA CAVA

TESTA

TYPE THE NAME OF YOUR TEST FILE.  ? 001
001 HAS 50 ITEMS, PROFESSOR.
PROFESSOR, COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE B
PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS!  ? 3
WHAT FILE QUESTION SHOULD I BEGIN WITH!  ? 11
WHAT PATTERN SHOULD I FOLLOW:  1 EVERY QUESTION
2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ? 2
PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER.
ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.?
```

```
                EXAMPLE B                                  1

1 . THE RELAXATION PHASE OF THE CARDIAC CYCLE IS CALLED!
        1 .   DIASTOLE
        2 .   REFRACTORY
        3 .   ARTERIOLE
        4 .   SYSTOLE

2 . THE DEVICE FOR MEASURING BLOOD PRESSURE IS!
        1 .   CARDIOMETER
        2 .   ELECTROCARDIOGRAPH
        3 .   STETHOSCOPE
        4 .   SPHYGMOMANOMETER

3 . THE BLOOD ENTERS THE RIGHT ATRIUM THROUGH:
        1 .   INFERIOR VENA CAVA
        2 .   DESCENDING AORTA
        3 .   SUPERIOR VENA CAVA
        4 .   ASCENDING AORTA
```

EXAMPLE B addressed the same file of test items as EXAMPLE A. The program, however, through a random process, changed the order of the answers, in effect preparing a second test form with the same quiz items. Additional forms, if run, would also differ.

Two example runs of TESTB, which produces true-false tests and answer keys, follow:

```
TESTB

TYPE THE NAME OF YOUR TEST FILE.  ? 001
001 HAS 58 ITEMS, PROFESSOR.
PROFESSOR, COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE C
PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS:  ? 3
WHAT FILE QUESTION SHOULD I BEGIN WITH:  ? 11
WHAT PATTERN SHOULD I FOLLOW!  1 EVERY QUESTION
2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ? 2
PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER.
ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.?

                 EXAMPLE C                           1

  1 . THE RELAXATION PHASE OF THE CARDIAC CYCLE
       IS CALLED SYSTOLE.                          1 . ------

  2 . THE DEVICE FOR MEASURING BLOOD PRESSURE
       IS SPHYGMOMANOMETER.                        2 . ------

  3 . THE BLOOD ENTERS THE RIGHT ATRIUM
       THROUGH INFERIOR VENA CAVA.                 3 . ------
```

```
TESTB

TYPE THE NAME OF YOUR TEST FILE.  ? 001
001 HAS 58 ITEMS, PROFESSOR.
PROFESSOR, COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE D
PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS:  ? 3
WHAT FILE QUESTION SHOULD I BEGIN WITH!  ? 11
WHAT PATTERN SHOULD I FOLLOW!  1 EVERY QUESTION
2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ? 2
PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER.
ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.?

                 EXAMPLE D                           1

  1 . THE RELAXATION PHASE OF THE CARDIAC CYCLE
       IS CALLED DIASTOLE.                         1 . ------

  2 . THE DEVICE FOR MEASURING BLOOD PRESSURE
       IS SPHYGMOMANOMETER.                        2 . ------

  3 . THE BLOOD ENTERS THE RIGHT ATRIUM
       THROUGH INFERIOR VENA CAVA.                 3 . ------
```

EXAMPLE C and EXAMPLE D addressed the same file of test items addressed by EXAMPLES A and B. Again, through a random process, two forms of the same quiz have been prepared with the same quiz items.

All four programs address terminal format files. A short example file for a bank of test items (shown with line numbers for convenience) follows:

```
100 3
110 AN,UNSTEADY HEARTBEAT,IS,ARRHYTHMIA,ECTOPIC
120 ISCHEMIA,BRADYCARDIA
130 THE,NORMAL HEART BEAT,IS,72 BEATS/MINUTE
140 82 BEATS/MINUTE,100 BEATS/MINUTE,60 BEATS/MINUTE
150 "",BLOOD ENTERS THE RIGHT VENTRICLE,THROUGH
160 TRICUSPID VALVE,MITRAL VALVE,BICUSPID VALVE
170 PULMONARY VALVE
```

To prepare a set of items for input to a terminal format file, the instructor may wish to write each question on an index card in a manner similar to this:

AN	(Determiner)
UNSTEADY HEARTBEAT	(Stem)
IS	(Verb)
ARRHYTHMIA	(Correct answer)
ECTOPIC	(Other answers)
ISCHEMIA	
BRADYCARDIA	

By using cards of this nature, the instructor can prepare an item or two at a time, setting aside his materials until he is ready to establish his test file online at the terminal. Cards of this type are particularly helpful where a senior or graduate assistant is to help input the items into the terminal format file.

In the terminal format file shown above it is noted that the first line of the file gives the number of test items within the file. The determiner for the first question is 'AN,' line 110. The determiner for the second question is 'THE,' line 130. In some instances a null determiner may be needed, as line 150, where the null determiner is represented by two double quote marks together without a space. In a print statement within the TEST programs this produces 'nothing.'

Since the determiners and verbs are set as separate data elements, TESTC, which produces matching items, may then address only the stems and correct responses to develop items like these:

```
TESTC

TYPE YOUR TEST FILE NAME.  ? 001
YOUR FILE HAS 50 ITEMS, HERR PROFESSOR.
COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE E
HOW MANY MATCHING SETS DO YOU WISH?  ? 1
AND HOW MANY ITEMS ARE TO BE IN EACH SET?  ? 5
BELOW YOU ARE TO ARRANGE THE 'ANSWER KEY' FOR THE
MATCHING SETS!
MATCHING SET  A!
     ITEM  1     ? 4
     ITEM  2     ? 3
     ITEM  3     ? 5
     ITEM  4     ? 1
     ITEM  5     ? 2

NOW ADJUST THE DITTO PAPER--AND THEN HIT THE RETURN KEY.?

                    EXAMPLE E                                          1

                 (A) PLEASE MATCH
  1 . UNSTEADY HEARTBEAT                                 BLOOD PRESSURE        -----
  2 . NORMAL HEART BEAT                                  TACHYCARDIA           -----
  3 . HEART RATE FASTER THAN 100 BEATS/MINUTE            TRICUSPID VALVE       -----
  4 . MEASURE OF FORCE OF BLOOD AGAINST A VESSEL WALL    ARRHYTHMIA            -----
  5 . BLOOD ENTERS THE RIGHT VENTRICLE                   72 BEATS/MINUTE       -----
```

In TESTC the instructor inputs the answer patterns at the beginning of the program run.

TESTD, which produces completion items, is rather simple and straight forward:

```
TESTD

TYPE THE NAME OF YOUR TEST FILE.  ? 001
001 HAS  50  ITEMS, PROFESSOR.
PROFESSOR, COMPOSE YOUR QUIZ TITLE!  ? EXAMPLE F
PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS!  ? 3
WHAT FILE QUESTION SHOULD I BEGIN WITH!  ? 11
WHAT PATTERN SHOULD I FOLLOW!  1 EVERY QUESTION
2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ? 2
PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER.
ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.?

                   EXAMPLE F                                  1

  1 . THE RELAXATION PHASE OF THE CARDIAC CYCLE
              IS CALLED                                1 .  -----------------

  2 . THE DEVICE FOR MEASURING BLOOD PRESSURE
              IS                                       2 .  -----------------

  3 . THE BLOOD ENTERS THE RIGHT ATRIUM
              THROUGH                                  3 .  -----------------
```

Terminal format files can be listed, sequenced, and desequenced. This allows the developer to build them over a period of time, if necessary. Additionally, because they can be listed at the terminal, they can be proofed and edited with ease. For those who have senior or graduate assistants with nimble fingers, the files can be developed rather quickly. Files are much easier to type than lines on a page of conventional typing. Once a file has been developed for a unit of instruction, in a matter of minutes, or an hour or so at the most, all of the quizzes or tests needed for the unit can be produced at one sitting, by executing the TEST programs you choose. The spirit masters can be safely filed away to be run when needed. For maximum security of the test files, they should be 'copied,' that is, 'punched,' on paper tape as soon as the test masters have been cut, and then the 'electronic' files can be 'unsaved.' Whenever the unit is given again, the paper tapes can be read back into the system and new test masters cut: the 'second time around' requires just minutes since the typing necessary to establish the file originally need not be repeated.

A listing of TESTA follows:

```
100' SJU    COLLEGE OF PHARMACY & ALLIED HEALTH PROFESSIONS
110'
120' DR. LOIS J. KELLER & R. L. WILLIAMS           9/22/77
130'
140' TESTA GENERATES MULTIPLE-CHOICE TESTS FROM TEST FILES
150'        ESTABLISHED  BY  THE INSTRUCTOR.  DOCUMENTATION
160'        OF TEST FILES MAY BE OBTAINED FROM THE PHARMACY
170'        EDUCATION RESOURCE CENTER...............EXT 673.
180'
190' ALSO SEE TESTB  FOR A  TRUE-FALSE  GENERATOR FROM THE
200'        SAME TEST FILES.  ALSO SEE TESTC & TESTD.
210'
220 MARGIN 132
230 DIM C(4),D(80),H$(4)
240 RANDOMIZE
250 PRINT "TYPE THE NAME OF YOUR TEST FILE.  ";
260 INPUT Y$
270 FILE #1:Y$
280 INPUT #1:S
290 PRINT Y$;" HAS";S;"ITEMS, PROFESSOR."
300 PRINT "PROFESSOR, COMPOSE YOUR QUIZ TITLE:  ";
310 LINPUT Z$
320 PRINT "PROFESSOR, GIVE THE NUMBER OF QUIZ ITEMS:  ";
330 INPUT T
340 PRINT "WHAT FILE QUESTION SHOULD I BEGIN WITH:  ";
350 INPUT Q
360 PRINT "WHAT PATTERN SHOULD I FOLLOW:  1 EVERY QUESTION"
370 PRINT "2 EVERY OTHER  3 EVERY THIRD  4 EVERY FOURTH  ";
380 INPUT R
390 LET R1=R
400 LET P=(T*R)+(Q-R)
410 IF P<=S THEN 460
420 PRINT "SORRY.  THIS WILL OVER RUN THE FILE.  LET'S"
430 PRINT "TRY AGAIN."
440 GO TO 320
450 PRINT
460 PRINT "PROFESSOR, NEXT YOU ARE TO ALIGN THE DITTO MASTER."
470 PRINT "ONCE THAT'S DONE YOU'LL NEED TO PRESS THE RETURN KEY.";
480 LINPUT D$
490 PRINT
500 PRINT TAB(20);Z$;TAB(65);"1"
510 PRINT
520 LET N=M=1
530 LET G1=0
540 FOR Z=1 TO S
550 INPUT #1:A$,B$,C$,D$,E$,F$,G$,
560 IF Z<>Q THEN 950
570 LET Q=Q+R1
580 LET I=INT(RND*4+1)
590 LET H$(I)=D$
600 LET D(N)=I
610 LET C(1)=I
620 LET J=INT(RND*4+1)
630 IF J=I THEN 620
640 LET H$(J)=E$
650 LET C(2)=J
660 LET K=INT(RND*4+1)
670 IF K=J THEN 660
680 IF K=I THEN 660
690 LET H$(K)=F$
700 LET C(3)=K
710 LET L=10-(I+J+K)
720 LET H$(L)=G$
730 LET C(4)=L
740 IF N>9 THEN 760
750 PRINT TAB(1);
760 PRINT N;". ";A$;" ";B$;" ";C$;"1"
770 FOR X=1 TO 4
780 FOR Y=1 TO 4
790 IF C(Y)=X THEN 810
800 NEXT Y
810 PRINT TAB(8);X;".  ";H$(X)
820 NEXT X
830 PRINT
840 LET R=R+R1
850 IF R>(R1*T) THEN 960
860 LET N=N+1
870 LET A=(N+G1)/11
880 IF A<>INT(A) THEN 950
890 LET M=M+1
900 FOR X=1 TO 4
910 PRINT
920 NEXT X
930 PRINT TAB(20);Z$;TAB(65);M
940 PRINT
950 NEXT Z
960 LET O=(M*65)-(N*6)
970 FOR X=1 TO O
980 PRINT
990 NEXT X
1000 PRINT "FOR THE ANSWER KEY, CLEAR THE DITTO MASTER,"
1010 PRINT "AND THE PRESS THE 'RETURN' KEY.  ";
1020 LINPUT D$
1030 PRINT
1040 PRINT TAB(20);Z$;TAB(60);"ANSWER KEY"
1050 PRINT
1060 FOR X=1 TO N
1070 IF X>9 THEN 1090
1080 PRINT TAB(1);
1090 PRINT X;".  ";D(X)
1100 PRINT
1110 NEXT X
1120 END
```

The TEST series is written in Dartmouth Time-Sharing BASIC. It would appear that only very minor alterations would be needed to convert the program to other versions of BASIC. Run time is minimal. Listings and documentation of the TEST series are available for the cost of handling from the Education Resource Center of the College by writing to the Dean, College of Pharmacy & Allied Health Professions, St. John's University, Jamaica, New York, 11439.

COMPUTER ASSISTED TESTING SERVICES

by

Dr. Hugh McHenry
Department of Mathematical Sciences
Memphis State University
Memphis, Tennessee

Computer Assisted Testing Services (CATS) is a set of BASIC computer programs designed to facilitate the use of multiple-choice tests in education, specifically in programs using individualized instruction. It is an interactive, time-sharing system operated by the instructor. CATS (1) Scores tests, (2) Stores test scores and answers, (3) Has numerous options for editing and managing records, (4) Provides for retrieval of data for students, instructors, and researchers, and (5) Generates test analyses. A special language containing 15 catagories of commands using conventional English words and syntax was created to execute the CATS programs.

Producing and grading tests are a major part of the instructor's job in most college courses. Therefore, carefull consideration should be given to this part of the teaching process when designing instructional programs. Attempts to do this in the Department of Mathematical Sciences at Memphis State University have resulted in using the computer to achieve three objectives in the testing programs in precalculus mathematics. In the first place, testing has been made an integral part of the learning process. Secondly, a systematic evaluation of the testing program is being made. Third, arrangements have been set up for the instructors to spend more time with their students on a one-to-one basis since they have been freed from most of the clerical chores of constructing tests and grading them, and keeping records. The system of computer programs used to achieve these objectives is described in this paper.

When it was decided that multiple-choice tests would be used, it was obvious that the computer could handle most of the clerical work. Therefore, an analysis was made of the tasks involved in grading multiple-choice tests and calculating the grades, and entering them in the grade book. The following seventeen tasks and/or teacher behaviors were identified as capabilities to be embodied in a set of computer programs.

1. Be operated by the instructor
2. Provide immediate results
3. Score Tests
4. Store students' records
5. Calculate grades
6. Prepare reports for students
7. Prepare reports for teachers
    a. Using names
    b. Using SSN
    c. Using grades
8. Analyze tests
9. Rescore tests
10. Allow for weighting
    a. Questions
    b. Distractors
    c. Tests
11. Allow changes to be made in the stored records
12. Handle several different tests in one set of papers
13. Accommodate several classes and/or courses at the same time
14. Permit addition and deletion of names and records in classes
15. Send messages to individuals and groups of students
16. Keep records of combinations of multiple-choice and essay tests
17. Create data files for research

These capabilities were developed in a set of programs called Computer Assisted Testing Services (CATS). A core image of the source program CATS1 is also called CATS and is used to simplify procedures for inexperienced users and to protect data files. CATS consists of the master control program CATS1 and a set of service programs which act as subroutines. Altogether there are 23 separately stored programs and 5 permanent data files, all of which are maintained on public disk storage. The programs are written in the BASIC language and require about 88,600 32 bit words of disk space for storage on the Xerox Sigma 9 computer. Experience with 608 students, each having 9 tests, indicates that about 170 words per student are needed for the data files.

The CATS package is an interactive, time-sharing system operated by the classroom teacher. A special language containing 15 catagories of commands and using conventional English words and syntax is used to obtain the services of CATS. All requests are issued in response to the system prompt of 'WHAT NOW?'.

TABLE 1 summarizes the conventions and terminology used in the CATS language. They are used to identify student groups, tests, computer files, and text strings. A student group may be as few as one person or it may be all the students in all of the courses on CATS. Each test is identified by a generic name such as TEST1A and by a four digit test code. The generic name enables students and instructors to identify a test while the code is used internally in CATS in scoring the test. Some of the services in CATS can be used on data stored in computer files. These services require that the file be specified by name. Text strings of up to 72 characters can be manipulated in the system. They may be messages to students or any other information on students records. Such strings may be inserted, edited, typed, replaced, or deleted for any student group.

A request for one of the CATS services is executed by entering the appropriate CATS command. TABLE 2 lists the possible commands. As indicated by the table, the commands have a uniform four-part structure. The first part is the

TABLE 1: Terminology and Conventions in the CATS Language

Term	Meaning
< > ---------	Enclose expressions representing a mandatory choice
[ ] ---------	Enclose expressions representing an optional choice
( ) ---------	Enclose information which does not constitute a CATS command
Note:	< >, [ ] and ( ) are used symbolically and are not typed when entering CATS commands
GROUP/ID ----	One of the following designations for a student group 1. ID for one student EG. 384-42-1288 2. Single student name EG. Coke, Larry 3. Course number EG. 1212 4. Course number and section EG. 1212 005 5. All (Means all students with the specified records)
CODE --------	A four digit number used to identify a test and its answer key in CATS EG. 3486
GENERIC NAME-	A six character name for a test which identifies the test for the student EG. TEST2F
TEST --------	One of the following designations for a test 1. Code [Generic name date ... etc] EG. 2785 TEST2B 2. Generic name [Date ... etc] EG. TEST5G 09/26/77
STRING ------	Any text string of up to 72 characters
FID ---------	The system symbolism for the name of a disk file
KEY#I -------	Line number I in a computer file

command verb which describes the task the user wishes to perform. Fifteen such verbs are listed in the command column in TABLE 2. These fifteen actions embody the tasks listed in the seventeen criteria cited at the beginning of this paper. The second part is called the object field and generally identifies what records are to be acted upon. Sometimes file names are used to identify the records while at other times tests, text strings, and line numbers in files are used. TABLE 2 lists 12 different object fields. The third part of the command structure is the word

TABLE 2: The Commands in Computer Assisted Testing Services

COMMAND	OBJECT FIELD	FOR	GROUP FIELD
ANALYZE	<TEST>	FOR	<GROUP/ID>
			<TEMP[ORARY]>
			<FILE>
BATCH	SUMMARY	FOR	<GROUP/ID>
BUILD	DATAFILE	FOR	<COURSE[SEC]>
BUILD	DIRECTORY		
COMPUTE	CUM[ULATIVE%]	FOR	<GROUP/ID>
DELETE	ANSWERKEY	FOR	<CODE>
	<STRING>	FOR	<GROUP/ID>
	<TEST>	FOR	<GROUP/ID>
	<KEY#I[-KEY#T]>		
EDIT	ANSWERKEY	FOR	<CODE>
	ROSTER	FOR	<GROUP/ID>
	<STRING>	FOR	<GROUP/ID>
	<TEST>	FOR	<GROUP/ID>
	<KEY#1[-KEY#T]>		
HELP			
INSERT	ROSTER		
	ANSWERKEY	FOR	<CODE>
	<STRING>	FOR	<GROUP/ID>
	<TEST>	FOR	<GROUP/ID>
	<KEY#I>		
REPLACE	<STRING1> WITH <STRING2>	FOR	<GROUP/ID>
RESCORE	<TEST>	FOR	GROUP/ID
SCORE	[DIRECTLY]	[FOR	<PERM[ANENT STORAGE]>]
			<TEMP[ORARY STORAGE]>]
	[FID]	[FOR	<PERM[ANENT STORAGE]>]
			<TEMP[ORARY STORAGE]>]
SUM[MARY]	<GROUP/ID>		
TYPE	DATAFILE		
	COURSE TEST SEQUENCE	FOR	<COURSE SECTION>
	ANSWERKEY	FOR	<TEST>
	ROSTER	FOR	<COURSE [SECTION I.D. ...ETC]>
	SUMMARY	FOR	<GROUP/ID>
	<STRING>	FOR	<GROUP/ID>
	<TEST>	FOR	<GROUP/ID>
	<KEY#I[-KEY#T]>		

(FEED THE ID SIDE OF A COMPLETED TEST FORM THROUGH THE 3-M READER TO:
1. DETERMINE IF THE TEST IS ON THE STUDENT'S FILE,
2. OBTAIN THE STUDENT'S GRADE ON THE TEST.)

(DEPRESS ESC KEY TWICE TO STOP THE CATS PROGRAM)

FOR. It is used as a separator for parts two and four and must be used when a command involves a fourth part. The last component in the formation of a command is called the group field. Generally it identifies the student group or test for which the command is to act. The command fields must be separated by at least one blank space to be interpreted.

The information provided the student and the instructor ultimately determines the value of the system. The reports prepared for individual students are the most common outputs from CATS. TABLE 3 contains two copies of the output of one student's record. The first line of each copy indicates that the record is for a student Jon P Sample, SSN 415-99-4226, in section 001 of course 1000. The lines beginning with KB2 identify the tests in the course 1000. Jon P Sample has taken four of the nine tests in the course. The grades are listed in the score column. The test identifiers are the generic names for the tests. For example, in KB201D KB is the author of the textbook, 2 is for the second edition of the text, 01 is chapter 1 and D is the form of the test. The symbols falling between the date and score for each test are the answers to the questions as supplied by the student. Each symbol is the answer to one question.

The top copy of the student's record in TABLE 3 is the kind produced by a time-sharing terminal. Students' answers on the tests are indicated by upper and lower case characters. Upper case characters indicate the student got maximum credit for those answers. A lower case symbol means that less than full credit, or no credit, was granted for the question. Printing such reports for 20 or more students on a 30 cps printer is not practical because of the time involved. Consequently, this method of output is used only for small numbers of students; Less than five is recommended.

TABLE 3: Samples of CATS Output of a Student's Record

```
1000 001 415-99-4226 SAMPLE, JON P
--------------------------------------------------------------------------------
WORK RECORD AS OF MAR 13 1978  CUMULATIVE COURSE SCORE:  38.5
--------------------------------------------------------------------------------
CODE RECORD  DATE   *SEE YOUR TEACHER TO CORRECT YOUR CODING ERRORS

CUMULATIVE% 03*13*78  86.55%
--------------------------------------------------------------------------------
TEST      DATE        INCORRECT ANSWERS INDICATED BY LOWER-CASE       SCORE
                  1...*...10....*...20....*...30....*...40....*...50  (%)
KB201D  01/25/78 UEkYJDEacDcBPBFd:kF\ICEDBEDAcCBdf{G:{PS                66.6
KB202E  02/03/78 BBAAERCXCKJH^DEWUx^B^]ZN\:TEGIEADKQBAABCa              95.1
KB203A  02/15/78 -KFHTRGKRCHTD}DQDBABARB                                95.5
KB204D  02/27/78 IECFPBNSXOweDBAPrLAfXAJYMPIOUL                         88.8
KB205
KB206
KB208
KB209
KB2FL
```

```
1000 001 415-99-4226 SAMPLE JON P
--------------------------------------------------------------------------------
WORK RECORD AS OF MAR 13 1978  CUMULATIVE COURSE SCORE:  38.5
--------------------------------------------------------------------------------
CODE RECORD   DATE   *SEE YOUR TEACHER TO CORRECT YOUR CODING ERRORS

CUMULATIVE% 30*13*78  86.55%
--------------------------------------------------------------------------------
TEST      DATE    INCORRECT ANSWERS REPLACE BY #(. IF OMITTED)        SCORE
                  1...1...10....1...20....1...30....1...40....1...50  (%)
KB201D  01/25/78 UE#YJDE##D#BPBE###FlICEDBEDA#CB###G##PS                66.6
KB202E  02/03/78 BBAAERCXCKJH:DEWU#:B: ZNl#TEGlEADKQBAABC#              95.1
KB203A  02/15/78 -KFHTRGKRCHTD#DQDBABARB                                95.5
KB204D  02/27/78 IECFPBNSXO##DBAP#lA#XAJYMPIOVL                         88.8
KB205
KB206
KB208
KB209
KB2FL
```

The bottom copy of the record in TABLE 3 is the kind produced for each student in class size groups. It is done on a line printer using batch processing and differs from the top copy only in the way the questions for which the student did not get maximum credit are indicated. Each such question has the symbol # inserted for the answer on the printout due to the fact that lower case symbols are not available on most line printers.

A look at test KB204D shows that four questions were "missed" on a test containing 30 questions. However, 26/30= 86.7% is not the grade reported since a system of weights assigned various values to the respective distractors for each question. The student picked up 88.8 - 86.7 = 2.1 percentage points by selecting distractors other than the maximum credit ones. That is, partial credit was given for some of the answers reported by the student. In general CATS allows weights in the interval (-999,999) to be assigned to tests, and/or questions and/or distractors. For example, each test in TABLE 3 has 11.11 weight points.

Two grades indicating the student's standing in the course are provided in addition to the individual test scores. The first is called the CUMULATIVE COURSE SCORE and is given in line two of the student's report (See TABLE 3). It is a number which will be 100 for the perfect student at the end of the course. Therefore, it gives the course percentage grade at any time during the semester a printout is obtained. In particular, it is the final grade at the end of the term. The other grade is called CUMULATIVE% and is the percentage grade on the combination of all tests up through the last one taken. See TABLE 3 for an example.

Further examination of TABLE 3 shows that some of the test answer sets contain nonalphabetic symbols. This is due to the ways in which CATS will allow the standard, five-option multiple-choice answer sheet to be marked. Any combination of the five positions may be marked. For example, the combination ABC is defined to be G. As a result, up to 32 distractors may be used for a given question. With only 26 letters in the alphabet, six additional pairs of symbols had to be utilized to reach the 32 needed--six for upper case symbols and six for lower case characters. Experience indicates that these special symbols should be used only after the letters have been exhausted.

The CATS package is currently running on the Xerox Sigma 9 computer at Memphis State University. A Texas Instruments Silent 700 thermal printer is interfaced with a Datronics 5500 test scoring machine to form the time-sharing terminal. The Datronics equipment is marketed by the 3M Company. An OpScan 17 optical reader can also be used and is serving as a backup reader.

CATS has been used for the past year in a beginning algebra course and has proven to a very effective tool for managing the classes. About 1,100 students have taken the course. They were taught by 18 different instructors in 27 sections. Thirteen of the instructors were graduate students in the department. Under these circumstances CATS has been used to do the following things.

1. To implement a uniform testing program-- Five forms of each of nine tests have been developed. Each instructor is required to use these in his classes.

2. To have more frequent tests -- Nine tests are given instead of the usual three or four. Such frequent testing has caused the students to be more conscientious in their work.

3. To use uniform grading standards -- All identical tests are graded by the same key and partial credit allocated in the same way for all students. The criteria for letter grades are derived from the data collected by CATS and based upon all students in the course. An A in one section means the same as in any other.

4. To keep students informed --The reports to the students provide a comprehensive statement of all the tests taken and the cumulative scores tell each student where he stands in the course.

5. To spend more time with students -- Since the teacher is not tied up as a clerk in testing, he is asked to man a learning center for the students. The center is open up to 20 hours per week.

6. To analyze tests -- Test analyses are used to identify problem areas and student weaknesses. Thus instructors know where to spend more time. Also, the analyses allow the tests to be carefully evaluated and revised.

7. To pace the course -- The departmental exams and uniform grading standards make it necessary for each instructor to see that his students cover all the prescribed course topics for each exam. Thus, he plans his work carefully.

CATS is being used to facilitate instruction and learning by creating an improved working environment for both the teacher and the student.

294

THE EVALUATION OF COMPUTER TESTING:
DESIGN STRATEGIES AND FEASIBILITY

Stuart D. Milner
School of Education
The Catholic University of America
Washington, D.C. 20064

and

A.M. Wildberger
School of Business and Public Administration
Howard University
Washington, D.C. 20059

Computers have been used in a variety of ways for test construction and analysis. The literature in this field mainly describes methodological approaches and logistical concerns. However, there is very little published information on the evaluation of the educational effectiveness of computer testing. This paper discusses evaluation requirements, strategies and practical problems in the context of a federally funded (Title IV-C) computer testing project now in the second of three years of operation.

The evaluation plan is presented along with relevant variables, student, teacher, and system performance objectives. Also included are research questions that have implications for computer testing in general. The rationale that led to the development of the strategies is discussed vis-a-vis the constraints of the evaluation setting.

INTRODUCTION

This paper discusses evaluation requirements, strategies, and practical problems in the context of a federally funded (Title IV-C) computer-based testing project now in the second of three years of operation. The project, which began in July 1976, is a cooperative effort in the public schools of three rural counties in Southside Virginia: Amelia, Nottoway, and Lunenberg. It has applied computer-generated tests and their automated scoring to the areas of mathematics, social sciences, and science in grades 4-12. It is being expanded this year to include language arts and reading. The project serves approximately 233 teachers and 7,277 students. Using the CYBER 172 computer at the University of Virginia in Charlottesville and leased word-processing and optical scanning equipment operating off-line, tests are generated and/or scored on request from teachers in the three counties. Along with the routine scoring, detailed statistical analyses of student performance and results for each individual test item are also being provided.

The principal focus of this project has been on developing competence rather than on optimizing the automated system. The system then serves as a medium and a catalyst by which the teachers increase their professional skills by preparing and evaluating the computerized tests and by using these tests to monitor instructional progress and to evaluate program effectiveness. The basic assumptions on which the project was undertaken are that the use of computerized testing promotes individual instruction in two ways:

1. It frees teachers from time-consuming clerical duties and thus affords greater opportunity for individual attention in the classroom.

2. It requires criterion-based tests that must be designed to meet individual needs. (Virginia Department of Education, 1977a.) Naturally, the official evaluation of the project needed to treat these assumptions, to the extent possible, as hypotheses to be tested.

At the time of the project's conception, six major goals were identified:

1. Implement an operational instructional management system based on computer-assisted test construction and the monitoring of student performance.

2. Raise the academic achievement scores of Project students in mathematics, social sciences, and science.

3. Train teachers in the principles of computer-managed instruction and improve their attitudes toward this instructional approach.

4. Train teachers in the principles of tests and measurements.

5. Reduce the amount of time necessary for the construction and administration of

tests.

6. Increase the amount of testing in the educational programs in science, mathematics, and the social sciences.

As previously mentioned, the project is being expanded into the language arts and English areas. Also, a seventh goal has been added:

7. Identify administration and guidance services which could be facilitated through computer processing and develop computer systems for making these services more efficient. (Amelia County School Board, 1977)

This last goal is clearly a reflection of the administration's concern for making the continuation of the CATC economically feasible after federal funds have been expended. The performance of multiple services, especially those for which clear cost savings can be shown, would contribute to the total cost-effectiveness of owning or leasing the equipment necessary to continue the CATC services.

While these goals have not changed appreciably, it has become increasingly more obvious that they are both interdependent and time-dependent. The training goals must be met before the testing program can be operational, and the goals relating to testing and management must be met before any significant increase in student achievement can be expected.

THE FIRST YEAR

The project officially started in July 1976, but the first six months were spent mostly in designing the system (i.e.: acquiring equipment, writing programs, etc.) and in developing the classification scheme for the test item banks. This time and effort was essential in that the whole structure of objectives on which the tests were to be based needed to represent accurately and in detail what was actually being taught in the schools of the three counties. The tests, while generated and scored by the computer, are intended to be criterion based, teacher designed and teacher specified, so as to be relevant to the day-to-day progress of each teacher's classes. This classification scheme was developed by teachers and project personnel working together. The practical effect of these preliminary requirements was that actual test generation for student use was delayed until Spring 1977.

In order to familiarize the teachers with the system and to provide items for the test banks, in-service training was held in the Spring and Summer of 1977 during which the teachers produced items and requested tests which were specific to the material they were currently covering in their classes. These tests were actually taken by their students, scored by the system, and the analysis returned to the teachers for them to revise items as needed. Naturally, some teachers continued to use the system regularly, while others rarely did so after completing the in-service course. Thus all of the first year's data was obtained during the Spring semester, most of it as a direct result of in-service training.

In one area, U.S. History, the teacher-generated items were supplemented by others acquired from another school district (Los Angeles). This area alone received a detailed evaluation in June 1977. A pretest/posttest comparison with a control group in a neighboring county showed a significant difference in the adjusted scores of treatment and control students in favor of the treatment group (i.e.: those receiving computer-based tests.) However, less than 50% of the students in the treatment group (and exactly 2/3 of their teachers) indicated a positive attitude toward the project when they were polled at the conclusion of the school year. Anecdotal evidence indicated some difficulty was experienced in matching the externally acquired items to the material actually being taught. There was also some feeling expressed that both students and teachers were less-than-willing participants in this rather formal experiment. (Amelia County Public Schools, 1977; Pedersen, 1977)

Item banks in other areas (such as math and science) were still too small and spotty for detailed analysis at the end of the first year, although by that time they contained over 16,000 teacher-generated items contributed by 35% of the target teachers. Even more teachers had submitted tests for scoring. Ninety-six percent of all teachers who used the system in any way indicated a positive attitude toward the project, despite the severe logistical problems of operating over a three county area, averaging 100 miles from Charlottesville.

In June 1977, the authors of this paper were asked to perform an independent analysis of alternative mechanizations (system designs) which included the possibility of a local computer installation. The desirability of providing at least local data communications was recognized; but the best that the project was able to support financially in the second year was a courier service.

All the evaluators, auditors, state and county administrators, and project personnel were in general agreement that the first year of the project had been largely developmental. It had not provided enough data to do any more than indicate promise. The evaluation had been forced to treat as a complete year what was really only half a year of rapidly increasing use of a system that was still under development. The rate of usage was too variable, data extended over too short a time, and was too heavily influenced by external factors (viz.: the in-service courses) to provide any formal statistical basis for believing that the rapidly rising usage rate at the end of Spring 1977 would be continued in the Fall.

In fact it did not continue. The second year appears to be repeating the first year's cycle of slow start and rapid finish.

THE SECOND YEAR

The state and county administrators are relying heavily for long range decision-making on the second year's evaluation. Their (and our) major concerns include:

1. the amount of usage of the system
2. its acceptance by teachers and administrators,
3. its effects on student and teacher performance.

Concurrently, an attempt is also being made to expand the subject matter to include language arts and English, and to greatly increase the number of available items in all banks. Workshops have been held regularly to acquaint all potential users with the system. Another in-service course is being offered in the second semester of 1977-78. It includes the writing and entry of new test items by the participants. Thus, this year's evaluation, like the first year's, must be made of an evolving system undergoing rapid change.

From the point of view of the Project Director, the County School Superintendents, and the Virginia Department of Education, the crucial question is whether or not computer-based test generation and scoring will be *perceived* as having a sufficiently important effect on the students' performance as to warrant the counties continuing to fund this service after the project's federal funds terminate in June 1979. This essentially subjective judgment is further complicated by the poor rural nature of the three counties, by the fact that funds for computer use must be drawn from the personnel payroll, and by the uncertainties as to who the real decision-makers will turn out to be. In this very practical and realistic situation, one consideration which the evaluators must address is: what outcome (or combination of outcomes) is most likely to elicit a firm decision?

This year the authors are internal evaluators, expected to perform formative as well as summative evaluation. To an extent, these roles conflict. In the formative role, it behooves us to detect any possible failures of the project operations to meet expectations and to recommend specific improvements. If we do this diligently we can expect to be performing a summative evaluation of a project which has been substantially modified over its history. To track teachers' reactions to each change in the system would be beyond the resources available and would probably only serve to antagonize them due to the additional questionnaires, etc. it would require. The strategies used for evaluation (to be discussed in subsequent sections) must somehow take all these factors in consideration.

THE EVALUATION PLAN

The overall evaluation model advocated by the Virginia State Department of Education for Title IV-C projects is presented in the Appendix. We focus on the performance objectives and implementation aspects in this paper.

The problems associated with this evaluation have much in common with other technology-based evaluations. First, the CATC system is constantly under development. Activities such as adding to and modifying the item banks go on continually, and they are inherently a part of the system. This creates a possibility of confounding implementation with project impact (Alderman, 1975).

Secondly, all students are eligible to make use of the system. Accordingly, traditional experimental designs that require randomization or matched control groups are difficult, if not impossible, to use. This evaluation, like the TICCIT evaluation (Alderman, 1975) is an empirical field study and not a designed experiment.

Finally, because of the innovative and experimental nature of the system, it may be too early to make definitive cost analyses (Grayson, 1972; Seidel, 1970). There is, however, data on some of the cost factors, and these data will affect decision making.

In the remaining sections, we will focus on strategies for empirical investigation, problems encountered in such efforts, and discussions of feasibility. The rationale and considerations that led to the development of the strategies are discussed vis-a-vis the practical constraints on this evaluation. Many of the specific evaluation objectives have relevance for any evaluation of computer testing (or other technological innovation) and are generic in nature.

OBJECTIVES FOR TEACHERS

A major concern of the decision makers is teacher utilization of, and attitudes towards, the testing system. Not only are the administrators interested in the level of overall usage compared with non-computer testing, but also in the patterns of use of the various parts of the system. As described earlier, several logistical, hardware, software and financial concerns exist. If certain non-essential aspects of the hardware or software are underutilized or even not preferred, they may be eliminated or at least modified.

Accordingly, we designed objectives requiring that a high percentage (80%) of the teachers in the three counties do the following: 1) submit at least one set of tests for automated scoring; 2) submit items to the central banks; 3) request computer generated tests; 4) make use of test-related statistics; and 5) demonstrate knowledge of salient features of the project. These objectives are particularly important in determining the real impact of the project on the teachers. As Charters and Jones (1973) have pointed out, without adequate teacher participation in a project, there may be the risk of evaluation of a "non-event."

We also requested that a determination be made of the uses of the system other than for routine test requesting and scoring (e.g., practice exercises.) This was in response to administrative concern for widening the scope of applications. Another objective was designed to determine if computer-based test generation and scoring required less time on the average than in previous or concurrent non-computerized practices.

The data collection for these objectives is accomplished in two ways. A cumulative set of records is being kept in the Project Director's office, and records are being generated by the computer system. To date, there have been no operational problems and the methods seem quite feasible.

We plan to provide the appropriate statistics documented by situational factors. For example,heavier use of computer testing is made at major examination periods.

Certain administrative and technical factors have already inhibited usage and must be dealt with in our analyses. Due to a number of administrative difficulties, an in-service teacher training course for sixty teachers was delayed until late January of this year. For a variety of reasons, several classes were cancelled subsequently. As a result, teachers' use of the system has been delayed. Since this will cause less usage, the analysis must take it into account.

Because of computer hardware and software constraints, lower case printing by the system is not currently possible. Although not all teachers in the target group consider this a serious constraint, the English teachers, who represent approximately 20% of the target group, have expressed the desire to have lower case printing available before using the test generation aspect of the system.

There are several ways to deal with these problems or ones of a similar type. First, one can compare usage while the problems exist with usage after they have been remedied. This can be done in a straightforward manner using both descriptive and inferential statistics. Second, one can analyze amount of usage based on total available items. Third, one can make projections of usage based on periods of maximal, or maximal possible, operation.

Because of the importance of teachers' attitudes towards the system, we designed a questionnaire to assess their attitudes toward the following: 1) procedures for obtaining tests; 2) usefulness of the system in the classroom setting; 3) overall quality of the items; and 4) the general concept of using computers in education. The teachers will also express their attitudes towards incentives for using the system and the perceived turn around times. We will relate statistically certain of these attitudes to performance variables.

OBJECTIVES FOR STUDENTS

Directly assessing the effectiveness of the computer testing on student performance is a difficult task. We propose to do this by determining whether or not increased test taking enhances performance. (The idea that students can learn from tests is intuitively obvious but needs empirical support.) Our rationale for this is based on the computer test feedback, as well as the greater frequency, precision, and individualization of testing than would be the case without computer support. In order to test this assertion, we stated the following objectives:

1) For those students who have taken at least one computer-scored test, the product moment correlation coefficent between the number of computer scored tests

taken and their score on the last such test will be significant at the .05 level. (This data is to be collected through the computer system itself.)

2) For those students who have taken at least two computer-scored tests, there will be a positive difference between their score on the last test taken and the mean of their score on the first test taken, with a significance level of .05 using the t-test. (This data is to be collected by the computer system itself.)

In order to assess whether or not students actually use the feedback, we stated that: Of those students who have had the opportunity to take at least three computer-based tests in the same subject area, at least 60% will have made use of the computer-generated analysis of their performance in preparing for the subsequent test(s) (as measured by self-report and validated by a test of knowledge of the contents of the analysis given to a small random sample of students). Indirect measures of the effectiveness will be discussed later.

Student attitudes towards the system are also to be assessed by means of a questionnaire designed by us.

CATC SYSTEM PERFORMANCE OBJECTIVES

Since the current evaluation is based on the second year of operation, and most of the major hardware and software efforts have been completed, the decision makers expressed an interest in a number of ongoing system related activities. The following objectives were stated in response to these concerns:

1) All students who are given computer-based tests will receive individual feedback analysis sheets. (Documented automatically by the system with validation by self-report of the students.)

2) At least 50% more teacher generated items will be available on the item banks by May 30, 1978 than were on the banks on July 1, 1977. (Documented by the system.)

3) The number of the tests submitted in April 1978 will be at least twice the number submitted in April 1977. (Documented by the system and by records in the Project Office.)

4) A random sample of 100 teacher generated items drawn from the banks in April 1978 will show no more than 10% which are not usable by reason of syntax, grammar, spelling, typographical errors, or internal contradictions.

The data will be gathered either from the computer system itself or from student self-report. The former should present no major problems. The latter we will attempt to obtain through a questionnaire.

PROJECT MONITORING

Additional data relating to usage of the data banks--test requests, specific item selection, items written/modified--is being gathered on a cumulative basis. This will assist us in providing information about ongoing activities. It will also give us insights into usage patterns so that better planning decisions can be made.

The computer costs associated with test generation and scoring, by resource, will be determined. That is, we will present information on the hardware and software costs associated with testing. In doing so, we expect to provide data on average cost per test. All of this will be important for determining: 1) whether the system is potentially cost effective; 2) whether it is financially feasible for these counties; and 3) whether a less expensive alternative exists (viz.: use of leased or purchased minicomputers). All of these data are available from the computer system itself. Therefore, there should be no problem with respect to system data collection.

USE OF A PSEUDO EXPERIMENTAL DESIGN

Recently, a number of researchers have proposed that measures of quantity of schooling or involvement in educational settings might have potential effects on instructional outcomes (Wiley & Harnischfeger, 1974; Karweit, 1976; Salomon & Clark, 1977). Both Wiley and Harnischfeger and Karweit investigated how quantity of schooling (e.g., time in school) effects achievement. Enough data exist to indicate that this may be the case.

In a prior study of computer assisted test construction based on this same project, Pedersen (1977) found some evidence that the number of test questions served as a significant predictor of posttest performance.

Salomon & Clark (1977) proposed a statistical pseudo-experimental design as an improved way to do media research in naturalistic settings. They indicate that the researcher should "investigate not only who was more and who was less affected, but also what caused the effect." As they state:

> The Pseudo-experimental Design (somewhat different from Shulman's Epidemiological Strategy) is based on the measurement of three kinds of indepen-

dent variables: relevant individual differences of the students involved, instructionally significant components of the program, and amount of student exposure to--or involvement in--the program, Students differ as to the amount of their exposure to, or involvement in, an instructional program thus, exposure is a continuous major independent variable. The purpose of this approach is, then, to examine the extent to which amount of exposure or involvement differentially affects students. Since, however, the program is analyzed into its significant components, one can also attempt to answer the question of what elements in the program affect individual learners.

The design makes use of the statistical procedure of multiple regression analysis. Background data are used as predictors in order to partial out initial differences among students and amount of exposure is entered in as the last predictor. As they indicate, the independent variable of exposure "has values distributed over a wide range from total non-exposure, through many levels of partial exposure, to total exposure." Thus, after background variables having been "partialed out" the "net" contribution of amount of exposure can be assessed.

Salomon & Clark reported a study in which this design was used. In that study, which dealt with the effects of viewing Sesame Street on performance by Israeli students, the variance due to amount of exposure accounted for an additional 4.3% to 18.3%

We have hypothesized that quantity of testing (like quantity of schooling and amount of exposure to instructional technology) may affect performance. Because of the potential ease of generating tests by computer, this may turn out to be a decisive factor in the use of computer testing.

In our case, we plan to use the statistical pseudo-experimental design in at least one subject matter area, eighth grade physical science. This cross sectional analysis will give us insights into the viability of the approach as well as the validity of our hypotheses.

Using the proportion, "items passed divided by total items passed and failed," as the dependent variable, we will attempt to determine its predictability using the Fall SRA science achievement test, the midterm and final grades, the number of computer tests given and the number of computer generated items presented as independent variables in a multiple regression analysis. The last two variables are the exposure-related variables in the design.

If our hypotheses is correct and amount of exposure is a significant predictor of performance, then there are differential effects due to the exposure-related variables and any aggregate or group data will only partially explain effects of the system because it would lump together students who received widely varying amounts of exposure. If amount of exposure is not a significant predictor, then we have a good reason to rely more heavily on the aggregate data. In any event, we believe that exposure-related variables need further exploration in the third year of operation.

Finally, we have described the evaluation design strategies and feasibility of a large scale CATC project. Even under constraining situations, evaluation is feasible. While our designs were created in response to these particular constraints, they are general in nature and applicable to many practical evaluations of computer-based instructional systems.

REFERENCES

Alderman, D. Evaluation of the TICCIT Computer-Assisted Instructional System. Princeton, N.J.: Educational Testing Service, 1975.

Amelia County Public Schools. Final Evaluation Report for Computer Managed Instruction. Amelia, Virginia: 1977.

Amelia County School Board. Title IV-C Self-Evaluation Report for Computer Managed Instruction Project. Amelia, Virginia: 1976.

Charters, W. W., Jr., and Jones, John E. On the risk of appraising non-events in program evaluation. Educational Researcher, 2, 11, 1973.

Grayson, L. Costs, benefits, effectiveness: challenge to educational technology. Science, 175, 1972, 1216-1222.

Karweit, N. Quantity of schooling: a major educational factor? Educational Researcher, 5, 2, 1976.

Pedersen, N.G. An Evaluation of the Effect of a Computer-Assisted Testing Program on Instruction in United States History. Unpublished doctoral dissertation, University of Virginia, 1977.

Salomon, G., and Clark, R. E. Reexamining the methodology of research on media and technology in education. Review of Educational Research, 47, 1, 1977.

Seidel, R. J., and Kopstein, F. F. Resource Allocations to Effect Operationally Useful CAI. HumRRO Profestional Paper 12-70. Alexandria, Virginia: HumRRO, April 1970.

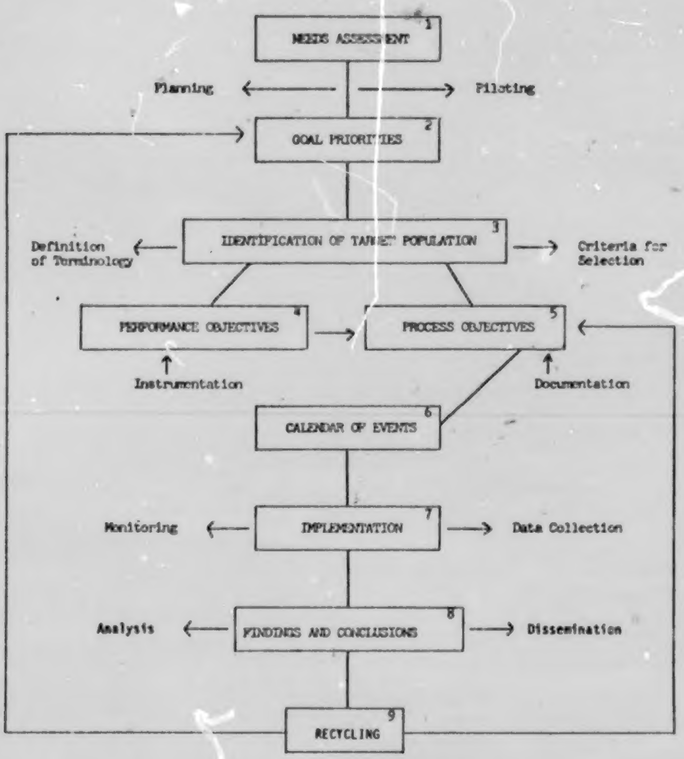
Virginia Department of Education. Profiles of ESEA, Title IV-C, Projects in Virginia, School Year 1977-1978. Richmond, Virginia: Federal Programs Office, October 1977a.

Virginia Department of Education. Guidelines for Evaluation. Title IV-C. Richmond, Virginia: Federal Programs Office, 1977b.

Virginia Department of Education. On-Site Evaluation Report for Computer-Managed Instruction Project. Richmond, Virginia: Federal Programs Office, 1977c.

Wiley, D. E., and Harnischfeger, A. Explosion of a myth: quantity of schooling and exposure to instruction, major educational vehicles. Educational Researcher, 3, 4, 1974.

APPENDIX


NEEDS ASSESSMENT 1
Planning
Piloting
GOAL PRIORITIES 2
IDENTIFICATION OF TARGET POPULATION 3
Definition of Terminology
Criteria for Selection
PERFORMANCE OBJECTIVES 4
PROCESS OBJECTIVES 5
Instrumentation
Documentation
CALENDAR OF EVENTS 6
Monitoring
IMPLEMENTATION 7
Data Collection
Analysis
FINDINGS AND CONCLUSIONS 8
Dissemination
RECYCLING 9

TITLE IV-C EVALUATION SCHEME
(from: Virginia Department of Education, 1977b, p. 7)

MEANINGFUL TEST-RESULT REPORTING

Viktors J. Muiznieks
Department of Secondary Education
University of Illinois, Urbana

Test-result reporting is one of the most neglected components in existing computer-assisted test construction (CATC) systems. Teachers are usually unprepared for, or uninterested in, interpreting the superabundance of numerical information produced by a typical test-grading program. This paper discusses some ways in which a computer program can be used to 'summarize' the statistical properties of an objective test so that teachers can receive verbal, descriptive statements about the general effectiveness of the test, about specific questions on the test, and even about particular alternatives within certain questions on the test. Although this paper focuses on a BASIC program written for an 8080 microcomputer system (CTRS), the principles discussed could be incorporated into almost any existing CATC system that performs the test-grading function.

Introduction

In the winter of 1976-77, an informal survey was conducted on the 100 largest public school systems and regional centers in the United States to learn about new formats and innovations in school test reporting (Burris). One of the questions asked was "What is your biggest unsolved need in test result reporting?" Burris found: "The answer to the question on the biggest unsolved need in test score reporting provided the most surprising portion of the survey. 42% of the responses to this item revolved around the lack of knowledgeable and effective use of test results. The situation seems to be that students, teachers, parents, etc., are not able to derive meaningful interpretations out of the array of numbers offered them -- numbers apparently with meaning only to statisticians and psychologists. ... Nor are breakthroughs apparent in helping teachers make better use of test results in day-to-day curriculum decisions." This paper will describe the computerized test reporting system (CTRS), which attempts to aid teachers in making more effective use of test results.

Ebel has stated "There is no better way for a teacher or professor to continue to improve his skills in testing, and the quality of the tests he uses, than to analyze systematically the results from tests and to compare the findings of these analyses with ideal standards of test quality." Unfortunately, teachers usually do not analyze their own tests, either because the task is too tedious and time-consuming, or because they do not know how. Remondini, for example, observes that without additional help, most teachers do not make a complete item analysis of the questions on every test because of the amount of time involved. Bloom, Hastings, and Madaus have found that "Teachers cannot reasonably be expected to have the time or statistical training necessary to perform test and item analysis. Further, many are justifiably not interested in acquiring this skill. Generally, therefore, teachers do not perform technical analyses of their own tests. However, a formal evaluation system demands that the analyses be made." With the use of a computer, teachers can obtain a complete item/test analysis for every test scored by the computer.

The item/test analysis information currently being provided by various test reporting systems is purely statistical in nature, and most teachers are not in a position to use such statistical results without additional explanation. CTRS is a test reporting system which uses item/test analysis statistics to produce verbal, descriptive statements which augment (and in some cases replace) numeric, statistical values so that "useful" results (which can be "meaningfully interpreted" by teachers) are produced.

CTRS Description

CTRS is an experimental system developed at the University of Illinois. It currently is being used by teachers in three nearby secondary schools. CTRS has been designed to help researchers study teachers' use of test results, and to examine the feasibility of designing an even more comprehensive CATC system built around the CTRS programs. If the test-result reporting mechanism is perceived as being useful by classroom teachers, then it makes sense to design a more elaborate CATC system which incorporates the features of item banking, item attribute banking, item generation, test specification, and test printing. If, however, teachers do not find the reported results useful, then the development of a more complex CATC system would be a waste of energy, effort, and resources. In any case, the general test-result reporting techniques used in CTRS could be incorporated into a number of existing CATC systems.

Since CTRS is being developed for use by classroom teachers, it was decided that the CTRS programs should be written in a computer language that is available to a large number of teachers. Therefore, CTRS programs are written in BASIC. Furthermore, since microcomputers are becoming so popular in secondary schools, CTRS has been developed on an 8080 microcomputer system.

Along with the 8080 mainframe, the CTRS programs require a serial port, a vector interrupt board, at least 32K but preferrably 48K of RAM memory, 2K of ROM memory, a mark-sense card reader with a specially-built interface, at least one but preferrably two floppy disk drives, and a paper-printing terminal. In addition, terminal ribbons, paper, floppy diskettes, and mark-sense cards must be supplied.

CTRS has been designed to require a minimum of teacher effort. Teachers simply leave the student answer cards (which are marked with a regular number 2 pencil), and an answer-key marked on a copy of the test, in a mailbox in the school's main office. The cards are picked-up after school every day, graded that night, and the results are returned to the teachers' mailboxes before school the very next day. CTRS allows for a variety of objective test-items to be graded, including multiple-choice, true-false, and matching.

There are two computer programs in the CTRS system. The first program is a file-preparation program which converts test-answers into a form acceptable to the second program (which performs the actual grading, analysis, and result-printing functions). The correct answers, all alternative names used in the questions, and all initial question difficulty estimates (if any) are typed-into the file-preparation program. The grading program then produces four printouts (see figures 1, 2, 3, and 4). Each of these printouts is described in greater detail in the next section.

CTRS Printed Reports

Figure 1 shows a sample printout of the student test-performance information and the overall test-analysis information. The student performance information includes the names of all the students taking the test (in alphabetic order), their scores, class rank, percentage correct, and question(s) missed. There is a provision for allowing the teacher to record a grade for each student on the printout itself. The overall test-analysis information includes the range of scores (as well as the high and low scores), the average score (rounded to the nearest integer), the Kuder-Richardson 21 test-reliability estimate adjusted to a 100-item test by the Spearman-Brown prophecy formula, the standard deviation, and the standard error of measurement (rounded to the nearest integer). Along with the test-reliability, standard deviation, and standard error of measurement values, the teacher receives a brief, descriptive explanation of what those values mean. In the case of the test-reliability estimate, the teacher receives one of three statements about the results of the test. These statements are that the results of the test probably are very reliable (if the estimate is greater than .75), reliable (if the estimate is between .4 and .75, inclusive), or somewhat questionable (if the estimate is less than .4). (Note that the number of statements which can be printed, the exact wording of the statements, and the range of the estimate values for each statement could easily be changed.) The teacher is also warned that reliability does not necessarily imply validity, although reliability is a necessary condition for validity.

Figure 2 shows a sample item-analysis printout. For each question on the test, the number of students choosing each alternative is displayed, along with the percentage of students answering each question correctly, the difficulty of each question, and an estimate of the question's discrimination ability. Three special symbols are used within the 'student response distribution' section. These are *(for indicating the correct answer to each question), + (for indicating an exceptionally good alternative--at least two-thirds of the students who chose the alternative ranked in the bottom-half of the class), and ? (for indicating a questionable alternative--students who ranked in the top-half of the class chose the alternative more often than students who ranked in the bottom-half of the class. The number and percentage of + and ? alternatives is printed at the bottom of the page along with the key describing how to interpret the symbols.

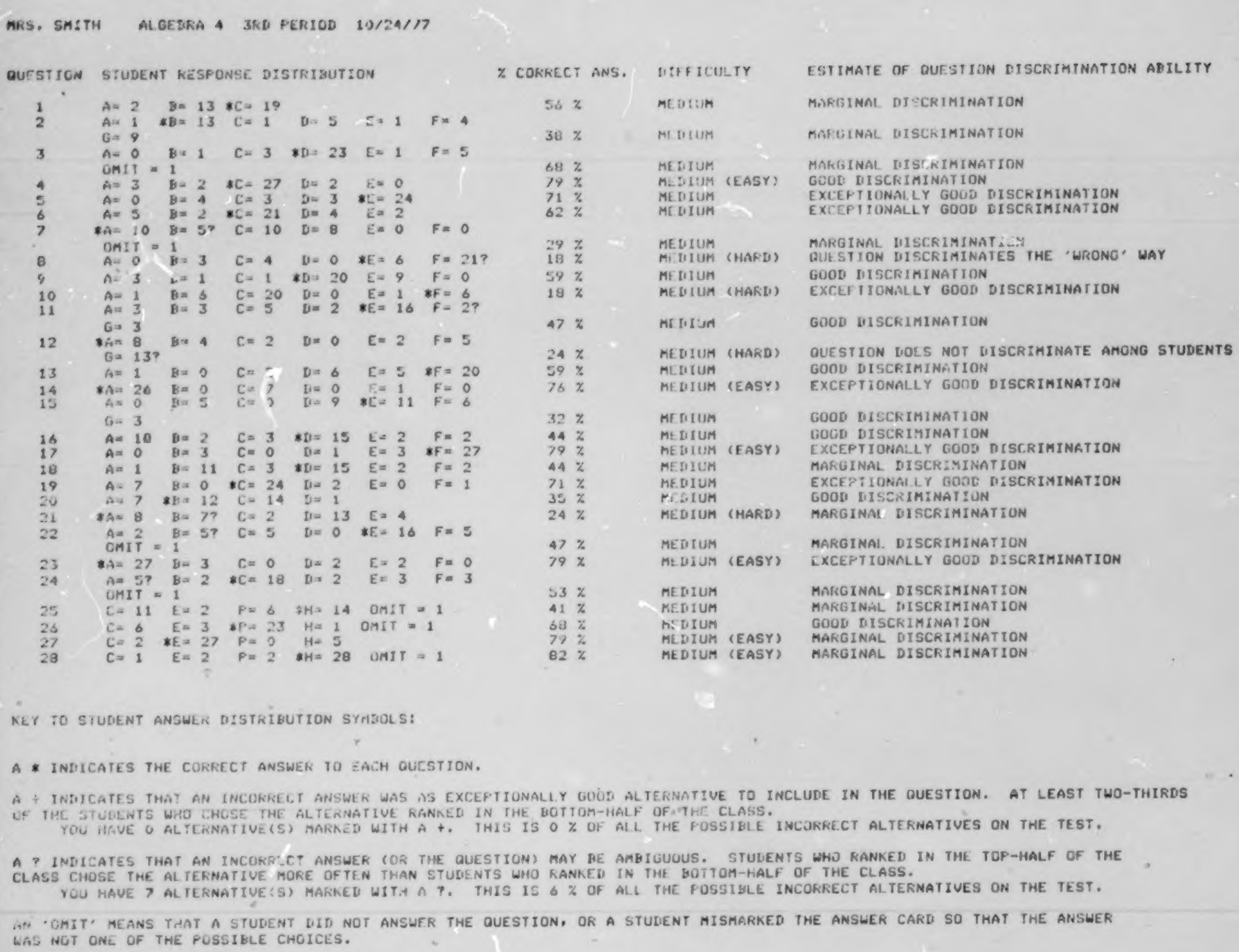
The difficulty of a question is either easy (if more than 75% of the students get the question right), medium (if 25-75% of the students get the question right), or hard (if less than 25% of the students get the question right). (Of course, the definitions of easy, medium, and hard could be changed; the number of categories could also be increased or decreased. The CTRS system uses the same difficulty assignments as used by Hazlett and Hofmann.) The difficulty column can contain one or two difficulty estimates. If the teacher exercises the option of estimating how hard each question will be before the test is given, then that estimate is printed first under the difficulty heading. If the teacher does not estimate the difficulty of each question, or if it turns out that the calculated difficulty is not the same as the estimate, then the calculated difficulty is printed in parentheses.

The estimate of discrimination ability is based on the item efficiency statistic described by Hofmann. This statistic is derived as a function of a question's difficulty and discrimination. Item efficiency is defined as the ratio of a question's observed discrimination to its maximum possible discrimination. (Hofmann demonstrates how the maximum discrimination of an item may be determined from its difficulty.) Hofmann also suggests the following method for interpreting the item efficiency statistic (e): if e is between 0 and .5, then the question is nonefficient; if e is between .5 and .8, then the question is efficient, and if e is greater than .8 then the question is ideally efficient. In CTRS, e is interpreted as follows: if e is greater than 0 but less than .5 then the question has marginal discrimination; if e is between .5 and .8, inclusive, then the question has good discrimination; if e is greater than .8 then the question has extremely good discrimination; if e=0 then the question does not discriminate between students, and if e is less than 0 then the question discriminates the 'wrong' way, i.e. the students who do poorly on the test tend to get the question right more often than students who do well on the test.

Figure 3 shows the item/test analysis summary printout where the teacher is told the number and percentage of hard, medium, and easy questions on

```
MRS. SMITH    ALGEBRA 4  3RD PERIOD  10/24/77

STUDENT NAME            SCORE   PERCENTAGE    RANK   GRADE    QUESTION(S) MISSED

ADAMS KENT                8        29 %        29    (   )    1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 22,
                                                              23, 24
ANDERSON BOB             10        36 %        24    (   )    2, 5, 6, 8, 9, 10, 11, 12, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24
ANDERSON PAUL            18        64 %        12    (   )    1, 2, 4, 8, 10, 12, 15, 18, 27, 28
BAKER ANGIE              11        39 %        23    (   )    2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 20, 21, 24, 25
BLACK STEPHANIE          19        68 %        8     (   )    1, 7, 8, 12, 15, 18, 20, 21, 22
BURNETT PAMELA           7         25 %        33    (   )    1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 20, 21, 22,
                                                              24, 25, 26
CLARK BILL               14        50 %        18    (   )    5, 6, 7, 8, 10, 12, 15, 18, 19, 20, 21, 24, 25, 26
DOUGLASS DALE            22        79 %        4     (   )    2, 7, 8, 10, 13, 21
GLENN JON                19        68 %        8     (   )    1, 2, 9, 10, 12, 15, 16, 21, 22
GREEN CRAIG              22        79 %        4     (   )    7, 8, 10, 12, 15, 25
HEART PAUL               10        36 %        24    (   )    1, 2, 6, 7, 9, 10, 11, 12, 13, 15, 16, 18, 21, 22, 23, 25, 26, 28
HEPBURN KATHY            8         29 %        29    (   )    1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 20, 21, 22, 23,
                                                              24, 25
JOHNSON DAVID            25        89 %        1     (   )    7, 8, 18
JONES JIM                23        82 %        2     (   )    8, 11, 12, 18, 25
JONES SHARON             23        82 %        2     (   )    7, 8, 12, 16, 18
KELLER ROY               9         32 %        28    (   )    2, 3, 6, 7, 9, 10, 11, 12, 14, 15, 16, 18, 19, 20, 21, 22, 24, 25,
                                                              27
LARSON MELANIE           17        61 %        13    (   )    1, 2, 8, 10, 12, 16, 18, 21, 22, 24, 25
MARCUS KEN               19        68 %        8     (   )    3, 8, 10, 12, 16, 20, 21, 22, 25
MARKS MICHELLE           10        36 %        24    (   )    2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18, 20, 21, 23, 25, 26
MILLER JAN               10        36 %        24    (   )    1, 2, 5, 6, 7, 8, 10, 11, 13, 15, 17, 18, 19, 20, 21, 22, 24, 25
NEUMANN BRAD             14        50 %        18    (   )    3, 4, 7, 10, 11, 12, 15, 16, 18, 20, 21, 22, 24, 26
PAINTER JOHN             20        71 %        7     (   )    7, 8, 10, 15, 21, 22, 24, 25
PETERSON LORI            16        57 %        15    (   )    1, 7, 8, 10, 12, 19, 20, 21, 22, 24, 25, 26
REID PEGGY               6         21 %        34    (   )    2, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24,
                                                              25, 26, 27, 28
ROBERTS HARRY            13        46 %        20    (   )    1, 2, 7, 8, 9, 10, 12, 13, 15, 17, 18, 19, 21, 23, 25
ROBINSON BARB            12        43 %        22    (   )    3, 5, 7, 10, 11, 12, 13, 14, 15, 16, 19, 20, 21, 24, 26, 27
ROLAND SCOTT             22        79 %        4     (   )    1, 3, 8, 12, 20, 24
SEARS RHONDA             13        46 %        20    (   )    1, 2, 6, 7, 8, 10, 11, 13, 16, 18, 19, 20, 21, 22, 26
SHAW MIKE                8         29 %        29    (   )    1, 3, 4, 6, 8, 9, 10, 11, 13, 14, 15, 16, 18, 19, 20, 22, 25, 26,
                                                              27, 28
SHEPHARD EDITH           19        68 %        8     (   )    2, 7, 8, 12, 15, 20, 21, 22, 26
SMITH CHERYL             8         29 %        29    (   )    2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20, 21, 25,
                                                              27, 28
WHITE JANICE             16        57 %        15    (   )    2, 5, 6, 7, 9, 10, 11, 12, 16, 21, 24, 25
WILLIAMSON FAY           16        57 %        15    (   )    1, 2, 7, 8, 9, 10, 11, 15, 20, 21, 27, 28
WRIGHT DAVID             17        61 %        13    (   )    2, 3, 8, 10, 11, 12, 13, 15, 20, 21, 25

THE TEST SCORES RANGED FROM 6 TO 25 --- A SPREAD OF 19 POINTS.

THE AVERAGE SCORE IS 15 POINTS.

THE THEORETICAL VALUE OF YOUR TEST RELIABILITY = .95 (OUT OF A POSSIBLE 1.00).  (THE AVERAGE RELIABILITY OF TEACHER-MADE TESTS IS
ABOUT .60).  THEREFORE, YOU PROBABLY CAN ASSUME THAT THE RESULTS OF YOUR TEST ARE VERY RELIABLE.
BEWARE!!!   A GOOD TEST SHOULD BE RELIABLE.  HOWEVER, JUST BECAUSE A TEST IS RELIABLE DOESN'T MAKE IT GOOD!
A VERY RELIABLE TEST COULD BE TESTING SOMETHING DIFFERENT FROM WHAT YOU WANT TO TEST.

THE SCORES ON THE TEST HAVE A STANDARD DEVIATION OF 5.41 (THIS MEANS THAT ABOUT TWO-THIRDS OF THE SCORES FALL BETWEEN 9 AND 20 ).
FROM THE STANDARD DEVIATION, AND THE TEST RELIABILITY, IT IS THEORETICALLY POSSIBLE TO CALCULATE HOW MANY POINTS EACH
STUDENT MAY HAVE MISSED (OR GOTTEN RIGHT) BY CHANCE FACTORS ALONE (SUCH AS GUESSING).  FOR THIS TEST, YOU CAN ASSUME
THAT EACH STUDENT'S SCORE IS CORRECT TO WITHIN 1 POINT(S).
```

FIGURE 1: SAMPLE PRINTOUT OF STUDENT TEST-PERFORMANCE INFORMATION AND OVERALL TEST-ANALYSIS RESULTS



```
MRS. SMITH    ALGEBRA 4  3RD PERIOD  10/24/77

QUESTION  STUDENT RESPONSE DISTRIBUTION                  % CORRECT ANS.   DIFFICULTY       ESTIMATE OF QUESTION DISCRIMINATION ABILITY

   1      A= 2   B= 13 *C= 19                                56 %         MEDIUM           MARGINAL DISCRIMINATION
   2      A= 1  *B= 13  C= 1   D= 5   E= 1   F= 4
          G= 9                                                38 %         MEDIUM           MARGINAL DISCRIMINATION
   3      A= 0   B= 1   C= 3  *D= 23  E= 1   F= 5
          OMIT = 1                                            68 %         MEDIUM           MARGINAL DISCRIMINATION
   4      A= 3   B= 2  *C= 27  D= 2   E= 0                    79 %         MEDIUM (EASY)    GOOD DISCRIMINATION
   5      A= 0   B= 4   C= 3   D= 3  *E= 24                   71 %         MEDIUM           EXCEPTIONALLY GOOD DISCRIMINATION
   6      A= 5   B= 2  *C= 21  D= 4   E= 2                    62 %         MEDIUM           EXCEPTIONALLY GOOD DISCRIMINATION
   7     *A= 10  B= 5?  C= 10  D= 8   E= 0   F= 0
          OMIT = 1                                            29 %         MEDIUM           MARGINAL DISCRIMINATION
   8      A= 0   B= 3   C= 4   D= 0  *E= 6   F= 21?           18 %         MEDIUM (HARD)    QUESTION DISCRIMINATES THE 'WRONG' WAY
   9      A= 3   B= 1   C= 1  *D= 20  E= 9   F= 0             59 %         MEDIUM           GOOD DISCRIMINATION
  10      A= 1   B= 6   C= 20  D= 0   E= 1  *F= 6             18 %         MEDIUM (HARD)    EXCEPTIONALLY GOOD DISCRIMINATION
  11      A= 3   B= 3   C= 5   D= 2  *E= 16  F= 2?
          G= 3                                                47 %         MEDIUM           GOOD DISCRIMINATION
  12     *A= 8   B= 4   C= 2   D= 0   E= 2   F= 5
          G= 13?                                              24 %         MEDIUM (HARD)    QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS
  13      A= 1   B= 0   C= 2   D= 6   E= 5  *F= 20            59 %         MEDIUM           GOOD DISCRIMINATION
  14     *A= 26  B= 0   C= 7   D= 0   E= 1   F= 0             76 %         MEDIUM (EASY)    EXCEPTIONALLY GOOD DISCRIMINATION
  15      A= 0   B= 5   C= 0   D= 9  *E= 11  F= 6
          G= 3                                                32 %         MEDIUM           GOOD DISCRIMINATION
  16      A= 10  B= 2   C= 3  *D= 15  E= 2   F= 2             44 %         MEDIUM           GOOD DISCRIMINATION
  17      A= 0   B= 3   C= 0   D= 1   E= 3  *F= 27            79 %         MEDIUM (EASY)    EXCEPTIONALLY GOOD DISCRIMINATION
  18      A= 1   B= 11  C= 3  *D= 15  E= 2   F= 2             44 %         MEDIUM           MARGINAL DISCRIMINATION
  19      A= 7   B= 0  *C= 24  D= 2   E= 0   F= 1             71 %         MEDIUM           EXCEPTIONALLY GOOD DISCRIMINATION
  20      A= 7  *B= 12  C= 14  D= 1                           35 %         MEDIUM           GOOD DISCRIMINATION
  21     *A= 8   B= 7?  C= 2   D= 13  E= 4                    24 %         MEDIUM (HARD)    MARGINAL DISCRIMINATION
  22      A= 2   B= 5?  C= 5   D= 0  *E= 16  F= 5
          OMIT = 1                                            47 %         MEDIUM           MARGINAL DISCRIMINATION
  23     *A= 27  B= 3   C= 0   D= 2   E= 2   F= 0             79 %         MEDIUM (EASY)    EXCEPTIONALLY GOOD DISCRIMINATION
  24      A= 5?  B= 2  *C= 18  D= 2   E= 3   F= 3
          OMIT = 1                                            53 %         MEDIUM           MARGINAL DISCRIMINATION
  25      C= 11  E= 2   P= 6  *H= 14  OMIT = 1                41 %         MEDIUM           MARGINAL DISCRIMINATION
  26      C= 6   E= 3  *P= 23  H= 1  OMIT = 1                 68 %         MEDIUM           GOOD DISCRIMINATION
  27      C= 2  *E= 27  P= 0   H= 5                           79 %         MEDIUM (EASY)    MARGINAL DISCRIMINATION
  28      C= 1   E= 2   P= 2  *H= 28  OMIT = 1                82 %         MEDIUM (EASY)    MARGINAL DISCRIMINATION


KEY TO STUDENT ANSWER DISTRIBUTION SYMBOLS:

A * INDICATES THE CORRECT ANSWER TO EACH QUESTION.

A + INDICATES THAT AN INCORRECT ANSWER WAS AS EXCEPTIONALLY GOOD ALTERNATIVE TO INCLUDE IN THE QUESTION.  AT LEAST TWO-THIRDS
OF THE STUDENTS WHO CHOSE THE ALTERNATIVE RANKED IN THE BOTTOM-HALF OF THE CLASS.
     YOU HAVE 0 ALTERNATIVE(S) MARKED WITH A +.  THIS IS 0 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.

A ? INDICATES THAT AN INCORRECT ANSWER (OR THE QUESTION) MAY BE AMBIGUOUS.  STUDENTS WHO RANKED IN THE TOP-HALF OF THE
CLASS CHOSE THE ALTERNATIVE MORE OFTEN THAN STUDENTS WHO RANKED IN THE BOTTOM-HALF OF THE CLASS.
     YOU HAVE 7 ALTERNATIVE(S) MARKED WITH A ?.  THIS IS 6 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.

AN 'OMIT' MEANS THAT A STUDENT DID NOT ANSWER THE QUESTION, OR A STUDENT MISMARKED THE ANSWER CARD SO THAT THE ANSWER
WAS NOT ONE OF THE POSSIBLE CHOICES.
```

FIGURE 2: SAMPLE ITEM-ANALYSIS PRINTOUT

```
OBSERVATIONS ABOUT YOUR TEST

     IF NO ONE CHOOSES AN INCORRECT ALTERNATIVE, THEN THAT ALTERNATIVE IS NOT VERY EFFECTIVE.  YOU HAD 24 INCORRECT
ALTERNATIVE(S) WHICH NO ONE CHOSE.  THIS IS 19 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.
      9 OF THESE WERE IN EASY QUESTIONS     11 OF THESE WERE IN MEDIUM QUESTIONS     4 OF THESE WERE IN HARD QUESTIONS
YOU SHOULD EXPECT TO HAVE SOME INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN EASY QUESTIONS.  HOWEVER, YOU PROBABLY DON'T
WANT A LARGE NUMBER OF INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN MEDIUM OR HARD QUESTIONS.

     THE DIFFICULTY COLUMN CONTAINS YOUR ESTIMATE OF HOW HARD YOU THOUGHT EACH QUESTION WOULD BE.  IF YOU DID NOT ESTIMATE
THE DIFFICULTY OF A QUESTION, OR IF IT TURNS OUT THAT THE CALCULATED DIFFICULTY IS NOT THE SAME AS YOUR ESTIMATE, THEN
THE CALCULATED DIFFICULTY IS PRINTED IN PARENTHESES.

     AN EASY QUESTION IS ONE WHICH MORE THAN 75% OF THE STUDENTS GET CORRECT.
     A MEDIUM QUESTION IS ONE WHICH 25-75% OF THE STUDENTS GET CORRECT.
     A HARD QUESTION IS ONE WHICH LESS THAN 25% OF THE STUDENTS GET CORRECT.

YOU ESTIMATED THE DIFFICULTY CORRECTLY ON 64 % OF THE QUESTIONS ON THE TEST.

     6 OF THE QUESTIONS ON YOUR TEST WERE EASY.  THIS IS 21 % OF ALL THE QUESTIONS ON THE TEST.
     18 OF THE QUESTIONS ON YOUR TEST WERE MEDIUM.  THIS IS 64 % OF ALL THE QUESTIONS ON THE TEST.
     4 OF THE QUESTIONS ON THE TEST WERE HARD.  THIS IS 14 % OF ALL THE QUESTIONS ON THE TEST.

     THE DISCRIMINATION OF EACH QUESTION IS BASED ON HOW WELL THE TOP-HALF OF THE STUDENTS DO ON THE QUESTION RELATIVE
TO THE BOTTOM-HALF OF THE STUDENTS.  THE DISCRIMINATION ESTIMATE IS NOT AN ESTIMATE OF HOW WELL THE QUESTION
MEASURES KNOWLEDGE, BUT RATHER OF HOW WELL THE QUESTION SEPARATES THE BETTER-SCORING STUDENTS FROM THE POORER-SCORING
STUDENTS.  THE ESTIMATE TAKES INTO ACCOUNT EACH QUESTION'S DIFFICULTY.  HOWEVER, THE ESTIMATE IS BASED ON
THE QUESTION AS A WHOLE, AND IT DOES NOT TAKE INTO ACCOUNT THE EXISTENCE OF ANY QUESTIONABLE INCORRECT ALTERNATIVES
(INDICATED BY ?'S).  THEREFORE, IT IS POSSIBLE TO GET AN 'EXCEPTIONALLY GOOD DISCRIMINATION' ESTIMATE AND STILL
HAVE SOME INCORRECT ALTERNATIVES MARKED WITH ?'S.

   'EXCEPTIONALLY GOOD DISCRIMINATION' MEANS THAT MOST OF THE STUDENTS WHO GOT THE QUESTION RIGHT WERE IN THE TOP-HALF OF THE CLASS.
        YOU HAD 7 QUESTION(S) MARKED AS HAVING EXCEPTIONALLY GOOD DISCRIMINATION.  THIS IS 25 % OF ALL THE QUESTIONS ON THE TEST.

   'GOOD DISCRIMINATION' MEANS THAT THE QUESTION DOES AN ADEQUATE JOB OF SEPARATING THE BETTER STUDENTS FROM THE POORER STUDENTS.
        YOU HAD 8 QUESTION(S) MARKED AS HAVING GOOD DISCRIMINATION.  THIS IS 29 % OF ALL THE QUESTIONS ON THE TEST.

   'MARGINAL DISCRIMINATION' MEANS THAT THE QUESTION JUST BARELY SEPARATES THE BETTER STUDENTS FROM THE POORER STUDENTS.
        YOU HAD 11 QUESTION(S) MARKED AS HAVING MARGINAL DISCRIMINATION.  THIS IS 39 % OF ALL THE QUESTIONS ON THE TEST.

   'DOES NOT DISCRIMINATE BETWEEN STUDENTS' MEANS THAT ALL STUDENTS DO EQUALLY WELL ON THAT QUESTION (REGARDLESS OF THEIR SCORE).
        YOU HAD 1 QUESTION(S) MARKED AS NOT DISCRIMINATING BETWEEN STUDENTS.  THIS IS 4 % OF ALL THE QUESTIONS ON THE TEST.

   'DISCRIMINATES IN THE WRONG DIRECTION' MEANS THAT THE POORER STUDENTS GOT THE QUESTION RIGHT MORE OFTEN THAN THE BETTER STUDENTS.
        YOU HAVE 1 QUESTION(S) MARKED AS DISCRIMINATING IN THE WRONG DIRECTION.  THIS IS 4 % OF ALL THE QUESTIONS ON THE TEST.

     IF A QUESTION DOES NOT DISCRIMINATE VERY WELL IT MAY STILL BE A VERY GOOD QUESTION.  IT IS POSSIBLE THAT THE QUESTION
MEASURES SOMETHING DIFFERENT FROM WHAT THE MAJORITY OF THE TEST MEASURES.  HOWEVER, IT MIGHT BE A GOOD IDEA TO LOOK AT
SUCH QUESTIONS TO SEE IF THERE MIGHT BE SOME PROBLEMS WITH THEM.  ANY QUESTIONS THAT DISCRIMINATE IN THE WRONG DIRECTION
SHOULD BE EXAMINED TO SEE IF THEY ARE AMBIGUOUS (OR IF THE ANSWER IS WRONG).

*** WARNING ***  BLIND ACCEPTANCE (OR THOUGHTLESS DISREGARD) OF THE ABOVE RESULTS COULD BE HAZARDOUS TO YOUR TESTING PRACTICES!!
     THE COMMENTS MADE ON THIS PRINTOUT ARE BASED ON (GENERALLY SOUND) TEST-ANALYSIS THEORIES.  THE PURPOSE OF TEST-ANALYSIS IS TO
POINT-OUT TO YOU THE POSSIBLE STRENGTHS AND WEAKNESSES OF YOUR TEST.  ALTHOUGH IT IS POSSIBLE TO MAKE GENERAL STATEMENTS ABOUT
YOUR TEST, YOU ARE THE ONLY ONE WHO KNOWS YOUR STUDENTS, YOUR CLASSROOM, AND THE PARTICULAR CIRCUMSTANCES UNDER WHICH THE TEST
WAS GIVEN.  THEREFORE, YOU MUST BE THE ONE TO DECIDE WHETHER OR NOT THE COMMENTS MADE ON THIS PRINTOUT ARE RELEVANT TO
YOUR PARTICULAR TEST.
     ALWAYS KEEP IN MIND THAT THE ABOVE RESULTS ARE NOT ABSOLUTE.  TESTS, AND TEST QUESTIONS, DO NOT HAVE UNALTERABLE
CHARACTERISTICS.  THE SAME TEST, GIVEN TO A DIFFERENT GROUP OF STUDENTS, OR EVEN TO THE SAME GROUP OF STUDENTS ON ANOTHER DAY,
COULD HAVE QUITE DIFFERENT RESULTS.  TEST-ANALYSIS IS NOT DESIGNED TO HELP YOU BUILD 'PERFECT' TESTS, BUT RATHER TO AID YOU IN
GAINING AN UNDERSTANDING OF THE TESTS YOU GIVE.  HOPEFULLY, YOU CAN USE THAT UNDERSTANDING TO IMPROVE YOUR FUTURE TESTS.
```

FIGURE 3: SAMPLE ITEM/TEST ANALYSIS RESULT SUMMARY

the test, and the number and percentage of questions in each of the five item efficiency categories. The teacher also is told how many alternatives were never chosen by a student in hard, easy, and medium questions. The teacher is given an explanation of the terms used on the printout, and some suggestions for interpreting the results. Also, the teacher is cautioned about placing too much emphasis on the analysis results.

All students receive a printout containing the test-heading information (the teacher name, class name, class period, and date), their name, the wrong answer (and correct answer) for each question missed, the total test score, and the percentage correct. For an example of a student's printout, see figure 4.

CTRS Report Interpretation

Bloom, Hastings, and Madaus point out that the computer alters the teacher's role from that of producer to that of consumer of test and item analysis data. The role of a computerized test-result reporting system, therefore, should be to facilitate the teacher's becoming an intelligent consumer of test and item analysis information.

In order to aid teachers to become intelligent consumers of item and test analysis information, it is essential that the goals of item and test analysis be understood. Anastasi claims that the purpose of item analysis in teacher-made tests is to identify deficiencies in the tests or in the teaching. Gronlund states that item analysis is usually designed to answer questions such as the following:

a) Did the item function as intended? (e.g. did norm-referenced test items adequately discriminate between high and low achievers?)
b) Were the test items of appropriate difficulty?
c) Were the test items free of defects?
d) Were each of the distractors effective?

The answers to these questions can lead to a number of benefits, including:

1) improving individual test items (in selecting or reviewing items for future use)
2) providing a basis for efficient class discussion of test results
3) providing a basis for remedial work
4) providing a basis for the general improvement of classroom instruction
5) increased skill in test construction.

By using CTRS printouts (see figures 1, 2, and 3), teachers can obtain some, or all, of the above listed benefits.

CTRS can be used to provide information which helps locate deficiencies in specific test items in several ways. The item analysis printout can be used to quickly identify both questions that do not discriminate between students, and questions that might be ambiguous to the better students. Also, questions with only marginal discrimination ability can be readily located. (Of course, just because a question does not discriminate well does not mean the question is bad; the question could very well be measuring something different from what the majority of the test is measuring.)

By looking for +'s and ?'s in the response distribution table, a teacher can locate extremely effective alternatives (marked with +'s), and questionable (possibly ambiguous) alternatives (marked with ?'s). If no one chooses an alternative, then that alternative is not very effective. The teacher is therefore informed of the number of alternatives that did not fool anyone in easy, medium, and hard questions. (It is expected that easy questions should have more incorrect alternatives which no one chooses than hard questions.)

CTRS can be used to provide information which makes discussion of test results more relevant. Easy items can be omitted, while difficult or ambiguous items can receive special emphasis. Incorrect alternatives which fooled a large number of students can also be discussed.

CTRS can be used as a basis for remedial work since particular areas of weakness can be discovered for each student, or the entire class. However, in order to gain this benefit the teacher needs to be able to relate the test questions to specific learning objectives. If a student (or the entire class) misses several questions covering a specific objective, this means that the student (or entire class) has not mastered that objective. The teacher can then choose to assign remedial work to specific students, or to the entire class.

If CTRS is used repeatedly for a course, or for several different sections of the same course, it is possible to obtain information which helps locate where teaching methods/materials need improvement. This information can be obtained by identifying material that is consistently mismatched with student performance. For example, material that is consistently too difficult for the students might suggest curriculum revisions or shifts in teaching emphasis. Similarly, errors in student thinking which persistently appear in item analysis data might direct attention to the need for more effective teaching procedures.

If a teacher chooses to exercise the option of estimating the difficulty of each question before the test is given, then the difference between expected question difficulties and actual question difficulties can be seen very easily. This information can be used during the discussion of class results, as well as in the evaluation of teaching techniques.

Item analysis reveals ambiguities, clues, ineffective alternatives, and other technical defects that were missed during the preparation of the test. This information can be used in the revision of test questions for future use. In addition, valuable experience in revising questions (such as rewording statements so that they are unambiguous and rewriting alternatives so that they are plausible) is obtained. By obtaining experience in reviewing and revising inadequate test items, teachers become more aware of the factors causing technical defects in test questions, and consequently they improve their test construction skills.

Summary

This paper has focused on the computer's ability to perform item and test analyses of teacher-made tests. It has discussed how the computer can be used to assist the teacher in obtaining, analyzing, and comparing item/test analysis information. However, as Bloom, Hastings, and Madaus point out, "the machine does not evaluate; this is a human responsibility. ... The machine takes over the clerical but not the judgemental aspect of the evaluation system." The responsibility of interpreting, evaluating, and judging the information provided by the computerized system still belongs to the human teacher. A major feature of the information provided by the CTRS programs is that the format for presenting the information uses descriptive statements along with numeric values. Hopefully, this presentation format will be easier for teachers to comprehend and use, and therefore be more meaningful.

Further information about the CTRS system may be obtained by writing to the author.

Bibliography

Anastasi, Anne, Psychological Testing (fourth ed.), Macmillan Publishing Company, Inc., 1976.

Bloom, Benjamin S., J. Thomas Hastings and George F. Madaus, Handbook on Formative and Summative Evaluation of Student Learning, McGraw-Hill, 1971.

Burris, Bradford, "Innovative Test Reporting Systems," Proceedings of the 15th Annual Convention of the Association for Educational Data Systems, 235-239, 1977.

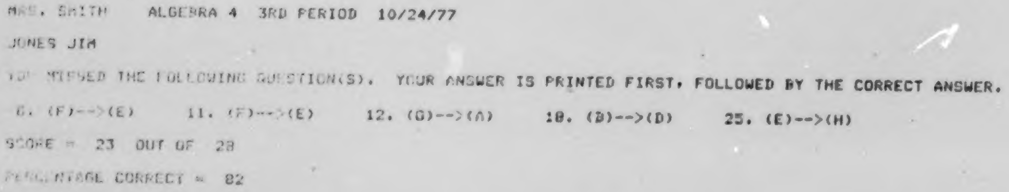
Gronlund, Norman E., Measurement and Evaluation in Teaching (third ed.), Macmillan Publishing Company, 1976.

Hazlett, C.B., "MEDSIRCH: A Computerized System for the Retrieval of Multiple Choice Items," Alberta University, Edmonton, Department of Educational Administration, Ottawa, Canada, Report Number DERS-3-70, August, 1970.

Hofmann, Richard J., "The Concept of Efficiency in Item Analysis," Educational and Psychological Measurement, Volume 35, Number 3, 621-640, Autumn, 1975.

Lippey, Gerald (ed.), Computer-Assisted Test Construction, Educational Technology Publications, 1974.

Remondini, David J., "Test Item System: A Method of Computer Assisted Test Assembly," Educational Technology, Volume 8, Number 3, 35-37, March, 1973.



```
MRS. SMITH    ALGEBRA 4  3RD PERIOD  10/24/77

JONES JIM

YOU MISSED THE FOLLOWING QUESTION(S).  YOUR ANSWER IS PRINTED FIRST, FOLLOWED BY THE CORRECT ANSWER.

 6. (F)-->(E)     11. (F)-->(E)     12. (G)-->(A)     18. (B)-->(D)     25. (E)-->(H)

SCORE =  23  OUT OF  28

PERCENTAGE CORRECT =  82
```

FIGURE 4: SAMPLE STUDENT PRINTOUT

COMPUTER-ASSISTED TEST CONSTRUCTION UNDER TITA2

P. L. Olympia, Jr.
Department of Chemistry
University of D.C. - Van Ness
Washington, D.C. 20008

An efficient procedure for the automated generation of repeatable tests that are as discipline-free as possible is described. With this procedure, repetitive generation of criterion-referenced, variable-format tests using the same compact data bank is possible. Test items requiring numerical solutions can be produced without the necessity of writing individual subroutines to compute the correct answer. Enhanced test items may also be presented. A test item classification system which increases the transportability of the test package, as well as its ease of use, is also described.

INTRODUCTION

Computer-assisted test construction (CATC) systems generally fall into two categories depending on the method of test construction:

- those in which the computer merely retrieves previously stored test items based on some specified test item selection criteria, and
- those in which the test item is primarily computer-formulated during execution time.

Most CATC systems are of the first type and perform what is commonly referred to as computer-assisted test assembly. Their main advantage is that they are relatively easy to design and implement. Their disadvantage is that one needs an enormous test item bank and computer storage capability in order to produce tests with some degree of repeatability. Repeatable tests are easily formulated from CATC systems of the second kind. Such systems also require considerably smaller data banks but the programming design is more involved because test item formulation must follow a carefully laid out algorithm.

This report describes a test construction system of the second kind that incorporates state-of-the-art techniques in computerized testing. It was originally developed in 1973 to support a broad-based personalized system of instruction (PSI) program in chemistry in an institution lacking the human resources from which to draw competent proctors. It is used presently at UDC-Van Ness and other colleges to support instruction in chemistry, physics, anatomy and physiology and respiratory therapy. The system, called TITA2 (Totally Iterative Testing and Analysis, version 2), is really a computer-managed instruction system written in Extended BASIC, at the heart of which is the test generating package which is the focus of this report. Since the test construction procedure here is primarily machine-directed, TITA2 is probably better described as one example of a "person-assisted test construction" system.

GENERAL FEATURES

A unique feature of TITA2 is that the test item does not exist in the data bank as such. Instead, the compact data bank consists of test item fragments which the computer joins together to form logical test questions. A typical multiple choice item consists of the following parts:

- keyphrase }
- statementphrase } question stem
- distractorphrase .

For example, an item that might read

A nitrogen atom has how many valence electrons?
a) 3 b) 4 c) 5 d) 6 e) 7

would have a keyphrase of "A nitrogen atom" and a statementphrase of "has how many valence electrons?". Each of the choices in the list, including the correct answer, is referred to as a distractorphrase so that the list of choices can be considered to be a small pool of distractorphrases. It is clear from this example that joining a keyphrase with a statementphrase produces the question stem.

The data bank consists of clusters of related keyphrases, statementphrases and distractorphrases which, taken together, constitute a data minipool. The above item could have come from a minipool such as that shown in Table 1. Thus, a multiple choice item is formulated by randomly selecting one member of the keyphrase pool and one member of the statementphrase pool, joining the two, then randomly selecting a specific number of distractorphrase pool members which always includes the correct answer. The list of choices is always randomly arranged. There are no limits on the number of pool members nor are there any

KEYPHRASE POOL	STATEMENTPHRASE POOL	DISTRACTORPHRASE POOL
A nitrogen atom	has how many electrons in the L-shell?	0
		3
		4
A hydrogen ion (H+)	has how many valence electrons?	5
		6
An oxygen atom		7
		8

TABLE 1. An example of a simple mini-pool

KEYPHRASE POOL	STATEMENTPHRASE POOL	DISTRACTORPHRASE POOL
Nitrogen	has how many valence electrons?	0
		1
Oxygen	has how many 2p electrons?	2
		3
Chlorine	needs how many more electrons to have a noble gas valence shell structure?	4
		5
Fluorine		6
		7
		8

TABLE 2. An example of a complex mini-pool

limits on the number of words constituting a given phrase so that an item consisting of several printed lines may be formulated.

This procedure permits considerable savings in computer storage space because phrases common to several potential test items, such as distractorphrase pool members, need to be stored only once. Also, since the computer constructs test items from fragments rather than simply retrieve complete items that were stored previously, a user, during execution time, has the option of specifying the number of distractors to be included in the list of choices. If the specified number exceeds the number of distractorphrase pool members the item will be formulated using all distractors in the pool.

The minipool just cited is an example of a simple minipool because all members of the statementphrase pool are essentially analogous, that is, the correct answer is independent of the statementphrase selected. However, in a typical TITA2 data bank complex minipools predominate over simple ones. In a complex minipool the correct answer depends not only on the choice of the keyphrase but of the statementphrase as well. An abbreviated example of a complex minipool is shown in Table 2. Here, as usual, the selection of members of the keyphrase, statementphrase and distractorphrase pools is fully automated and random. The user exercises some control over which minipools are to be included in the search and eventual formulation of the desired test item but the machine has complete freedom in selecting and assembling related phrases.

OTHER OBJECTIVE ITEMS

Additional flexibility, using the same data bank from which multiple choice items are formulated, is achieved by taking advantage of the fact that a multiple choice question can be recast rather easily both as a true/false or completion item. In the former, the item is false if any of the distractors in the distractorphrase pool is used to complete the question and true if the correct answer is used instead. Since the probability of formulating a "true" test item is comparatively small (1/D where D is the number of distractorphrase pool members), students are asked to either correct the item or specify what is wrong with it whenever his answer is "false".

To recast a multiple choice item into a completion type item, the computer joins the keyphrase and statementphrase selected and prints a series of underlines (the "blank") where the distractorphrase (specifically, the correct answer) is supposed to go.

To formulate matching-type items from the same data bank, the computer gathers members of a simple minipool. The items than consist of matching members of the keyphrase pool with members of the distractorphrase pool. The statementphrase pool members are not used in the items per se but the first member is incorporated in the text of explanation that precedes the set of matching-type items. Since members of the statementphrase pool are effectively discarded here, complex minipools are not used to formulate matching-type items.

It is clear that TITA2's algorithm allows the computer to formulate a test item in any one of several different types using the same compact data bank - a distinct advantage over a scheme which simply retrieves preformed items.

ITEMS REQUIRING NUMERICAL COMPUTATIONS

To further insure the repeatability of generated tests TITA2 can formulate test items requiring numerical answers where numerical data are randomly generated at execution time. The machine computes the correct answer from a concisely coded formula so that it is not necessary to write a separate subroutine for each item. Here, the data bank builder provides an item skeleton where a pound sign (#) appears whenever a number is to be randomly generated. For example,

> A #M solution of nonelectrolyte at #°C exhibits an osmotic pressure of (in atm).

This item skeleton is banked with four numbers defining the upper and lower bounds for the two randomly generated data. A coded formula is also supplied which the machine uses to compute the correct answer, e.g.,

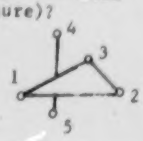
> 1*(2+273.)*0.0821.

In the formula, fixed point numbers are numeric constants which are part of the formula. Generated items such as this still may be cast as multiple choice, true/false or fill-in-the-blank.

ITEMS KEYED TO SPECIFIC DIAGRAMS

Some test questions require the presentation of a graph or figure. Traditionally, such "enhanced" test items are handled in CATC by providing the students with a booklet of uniquely numbered figures. For items requiring very elaborate figures this procedure is probably the only alternative given today's state-of-the-art unless one can administer the test interactively using a graphics terminal. However, for fairly simple diagrams one can store a set of point coordinates or a matrix of points along with the keyphrase. As in Denney's procedure (1) the student is then instructed to connect specific points to evolve the desired illustration. The example below shows an item keyed to a figure that is student-drawn using points provided by the machine.

> In the SCl4 molecule, according to VSEPR theory, the unshared electron pair should occupy which position (see figure)?
>
> a) 4
> b) 5
> c) either 4 or 5
> d) any of positions 1,2,3
> e) any of positions 1,2,4



ITEM CLASSIFICATION SYSTEM

A significant improvement regarding transportability of the test package comes about from the revised system of test item classification. Since TITA2 was conceived to support a competency-based education program its test generator component was designed primarily to produce criterion-referenced tests. Any potential test item is classifiable therefore under at least the two categories of terminal (major) and enabling objectives. To these we add another classification category that reflects the cognitive level, a la Bloom (2) of the behavioral objective which the potential test item purports to measure. The cognitive level assignment is used as an indicator of test item difficulty. In view of the widespread appeal of Bloom's taxonomy and its built-in hierarchy, using the cognitive level index as an indicator of item difficulty is deemed preferable to any local and arbitrary test item difficulty designation of "easy", "medium", "difficult".

A fourth classification dimension is used to indicate the particular subject matter or discipline to which a test item (and the behavioral objective the attainment of which it is designed to measure) traditionally belongs. Here, we use the 1000 divisions of the Dewey Decimal Classification (DDC) system which, although subject to occasional revisions, is nonetheless preferable to a less-known discipline code.

A given keyphrase is banked with its classification code which is a nine-digit number where the last three digits are separated from the first six by a decimal point placed there for programming convenience and to improve readability, that is ,

> DDDTTT.EEB

where DDD is the Dewey code in 1000 divisions; TTT is the terminal objective number; EE is the enabling objective number and B is Bloom'staxonomy level in the cognitive domani. Thus, a keyphrase with a classification code of 547015.183 is expected to appear in a generated test in organic chemistry (547) designed to measure the attainment of terminal objective 15 and enabling objective 18. The highest cognitive process involved for this item is one of application (level 3).

From the above description of the test item classification system it is clear that this could just as well be a classification system for behavioral objectives. That, of course, is the intent.

One obvious advantage of this classification system is that it allows for a straightforward test specification procedure using item attributes with which the average teacher is familiar. Also, since the compact data bank is keyed to a set of behavioral objectives it facilitates the exchange of data banks among users. Since the user can command the computer to generate a test item by citing as specific an attribute as an enabling objective it also avoids the problem common to simple computerized item retrieval systems where the computer occasionally retrieves an unwanted item.

For now a major drawback of an item classification system such as this is the dearth in published behavioral objectives. Our experience during the last four years has convinced us that identifying, writing and sequencing objectives for just a single discipline already constritute a gigantic undertaking for one institution.

EFFICIENCY OF TEST GENERATION

TITA2 produces a specified number of equivalent tests within a short time span which is essentially independent of the size of the data bank. This is accomplished, just prior to test generation, by the system's ability to create a renumbered and repacked subset of the original data bank containing only those potential test item fragments that satisfy the selection criteria specified. The randomized search for related test item phrases is then confined only to this subset. As the data bank undergoes revision the number of keyphrases that satisfy a given item characteristic naturally changes. When the system creates the data bank subset this information is gathered and given to the user so that he knows how many potential test items are in the bank that satisfy the desired characteristics. Consequently, even if the data bank undergoes daily update or revisions the user is assured of getting up-to-date information about the bank.

If the user does not have a list of desired behavioral objectives TITA2 will print one on request. The user may generate a list of objectives by indicating any combination of the four classification categories. He may also generate a list within a given DDC code by indicating index terms such as "pH". Since TITA2 not only formulates test items itself but also creates the data bank by prompting the terminal user for appropriate information, the entire test-making operation is essentially machine-directed.

CONCLUDING REMARKS

One of the painful realities of CATC is that it costs considerably more to build and update a high quality data bank of test items or item prototypes than it does to write computer programs that will use these data banks. Consequently, there is substantial advantage in encouraging users to exchange test items. Of course, one major problem that exists today is the lack of standards for test item classification that will permit a data bank built for a given CATC system to be used by another using a different system. The test item classification system used by TITA2 represents one attempt to somehow evolve a classification system that would be acceptable to most users. TITA2 users who wish to include other item attributes are free to do so provided that these attributes are assembled in another numeric code or line of code. It is essential to remember that the typical CATC user does not want to get bogged down in a quagmire of confusing and unfamiliar classification indices. There seems to be little logic in providing as many as 55 item attributes when the typical user tends to use no more than five of these.

Some disciplines such as mathematics and chemistry which routinely use special symbols present what at first might be considered problems in printing computer-generated tests. In this regard replaceable typing elements and special printing chains have been used. We, ourselves, have used neither. Our experience confirms what other CATC workers have noted: students readily adapt to such computer "idiosyncrasies" as writing chemical formulas with all letters capitalized or writing what would have been numeric subscripts on the same line with chemical symbols.

Although CATC has made impressive strides during the last five years much still remain to be done. Of these, perhaps the most significant would be the development of a special computer language for CATC just as there are now special tutorial languages to permit delivery of computer-assisted instruction courseware. As far as we know all CATC systems are wriiten in either all-purpose languages or languages designed primarily for other purposes. These languages are necessarily less efficient than one designed specifically for CATC. Millman (3) has attempted to design a CATC subset language within BASIC but its efficiency is hampered by the fact that it is subject to the syntax and limitations of the host language.

REFERENCES

(1) Denney, C., EDUC. TECH., 13 (3), 34 (1973).

(2) Bloom, B.S., "Taxonomy of Educational Objectives: Handbook I: Cognitive Domain", Longmans, Green and Co., New York, 1956.

(3) Millman, J. and Outlaw, W.S., "Testing by Computer", Cooperative Ext. Publ., Cornell Univ., Ithaca, 1977.

A NON-HIERARCHICAL COMPUTERIZED TEST ITEM DATA BASE

Irvin Rubincam and William P. Olivier
Department of Measurement, Evaluation, and Computer Applications
The Ontario Institute for Studies in Education
Toronto, Ontario, Canada M5S 1V6

ABSTRACT: A generalizable system of item classification and retrieval is described which will accommodate different curriculum philosophies. System produced tests are given to groups and interactively to individuals. Items are coded by multiple selections from keyword lists. Each keyword selected is of equal weight and independent of all other keywords. Weighting and dependence between keywords is introduced when retrieving items. Users specify keywords of interest, primary interest, excluded keywords, and the minimum number of keyword matches which must occur for item retrieval. Three Ontario nursing schools with differing curricula are using a common health science itembank. Other banks are in preparation.

Introduction

Access to a bank of test items under computer control can be of significant assistance to instructors in evaluating student progress. The usefulness of a particular testbank to a given instructor depends upon a number of factors including the appropriateness of the bank for the subject being taught and the ease and flexibility with which the item information can be retrieved.

The development of itembanks and supporting software has largely been in response to the needs of particular individuals, groups, or departments within single institutions. Decisions concerning how the items are to be categorized for inclusion in a bank are made within this context and generally reflect the specific curriculum philosophy of the designers. This focus on local needs is one of the main reasons that the transfer of itembank systems to sites other than the one for which they were created is not a common occurrence. Since it is wasteful of time and money to create a new system for each application, attention should be given to the development of generalizable procedures for item classification and retrieval which will accommodate a variety of curriculum philosophies in diverse educational settings.

This paper discusses one approach to this problem of generalizability and describes aspects of an interactive computer-assisted test construction system under development at the Ontario Institute for Studies in Education through funding provided by Seneca College of Applied Arts and Technology in Toronto. The initial itembank has been developed by the faculty of the Seneca College School of Nursing in cooperation with nursing schools at George Brown College in Toronto and St. Clair College in Windsor, Ontario. These three nursing schools each have their own way of structuring the health science curriculum. These differences had to be taken into account in the design of the coding and retrieval scheme.

Non-hierarchical Classification

In the initial stages of systems development it became clear that any attempt at a hierarchical classification scheme would bias the coding in favor of one curriculum structure or another since there is a lack of general agreement concerning the ideal organization of categories, subcategories, etc., in the health sciences. The challenge then was to develop a non-hierarchical classification scheme which would retain the advantages of a hierarchical scheme, namely the ability to widen or narrow the search parameters as circumstances demand, while eliminating the disadvantage of requiring classifications which may be inappropriate for certain users. In the scheme which was adopted items are coded by simply selecting keywords which match the content of the item. Each keyword which is selected is considered independent from and of equal weight with all other keywords. Weighting and the relationship between categories are introduced when items are retrieved from the bank. Users specify keywords of general interest, keywords of primary interest, excluded keywords, and the minimum number of keyword matches which must occur for an item to be retrieved. How these retrieval procedures operate in practice can best be seen by using a simplified keyword list as a source of examples.

Sample Keyword List

Consider an itembank which contains questions on food and nutrition. The keyword list contains the following terms:

Nutrition

1. Balanced diet - adults
2. Balanced diet - children
3. Nutritional deficiencies
4. Vitamin deficiencies

Families of Foods

5. Fruits
6. Vegetables
7. Meats
8. Dairy products

Foods

9. Milk
10. Eggs
11. Oranges
12. Apples

Nutrients

13. Protein
14. Vitamins
15. Vitamin C
16. Minerals
17. Fats

Sample Questions

The following two examples illustrate the type of question to be found in this itembank. The correct answer is marked with an 'X'.

ITEM NO. 3323

The nutrient that should be avoided in large amounts in a young child's diet is

(A) protein
(B) carbohydrates
(C) vitamins
(D) X fats

Keywords: balanced diet - children (2), fats (17)

ITEM NO. 5011

To increase Vitamin C intake most effectively you would select 1/2 cup of fresh juice of

(A) pinapples
(B) X oranges
(C) apples
(D) tomatoes

Keywords: oranges (11), Vitamin C (15)

Curriculum Structure and Itembank Coding

It is possible to specify some hierarchical relationships within the keyword list for this itembank. For example, 'apples' and 'oranges' are each a subset of 'fruits', and 'Vitamin C' is a subset of 'vitamins'. Even so, it would not be easy to produce a hierarchical arrangement of all of the keywords on this list. For example, should 'oranges' be a subset of 'Vitamin C' because oranges are an example of a food containing Vitamin C, or should 'Vitamin C' be a subset of 'oranges' because Vitamin C is one of the nutrients contained within oranges? Whenever decisions are made with regard to a hierarchical classification, other possibilties must be excluded. This is not to argue that a hierarchical structure should be avoided in a curriculum; quite the contrary, such a structure may be essential in conceptualizing a body of subject matter. But in an itembank each item may be considered a separate and independent measure of a person's ability in a particular discipline. If this is true, then it is important that the independence of each item be maintained in the coding process. The proper time to consider the interrelationships between items is when they are retrieved from the bank.

Item Retrieval

To retrieve items from the bank the instructor types a series of commands at a computer terminal. It usually takes the computer about 10 to 30 seconds to scan the contents of the itembank and inform the instructor of the number of items which have been found. These items can then be displayed one at a time and the instructor can select those which are to be included in a test.

Venn diagrams will be used to illustrate these retrieval procedures. The entire bank of items will be represented by the space within a rectangle. The items coded by each of the various keywords will be represented by-the space within circles. In those cases where items are shared by more than one keyword, an overlap between circles will represent the common items. These features are shown in Figure 1.

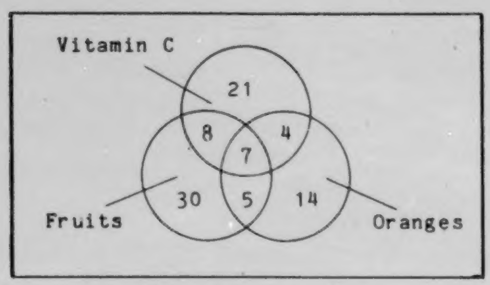


Fig. 1 - Sample item-keyword combinations

It can be seen that some items share two or three keywords. For example, 'fruits' and 'Vitamin C' share 15 items (8 + 7). Seven items are coded with all three keywords, while other items share no common keywords (30 items for 'fruits', 21 for 'Vitamin C', etc.).

The scheme used to select items from the itembank should be able to differentiate among the various areas shown in Figure 1. This is accomplished by two types of commands: (1) a list of keyword numbers, and (2) a specification of the required number of keyword matches (referred to as a 'match index').

The keyword number command can take one of three forms:

(1) Conditional Inclusion (Without a '+' or '-' Prefix) Items coded with a conditional keyword number will only be selected if the match index specifications are met.

(2) Absolute Inclusion (+)
A keyword number preceded by a '+' indicates that items must be coded with this keyword in order to be selected.

(3) Absolute Exclusion (-)
A keyword number preceded by a '-' indicates that items coded with this keyword will not be selected.

The match index command determines the number of keyword matches which must occur before an item will be retrieved.

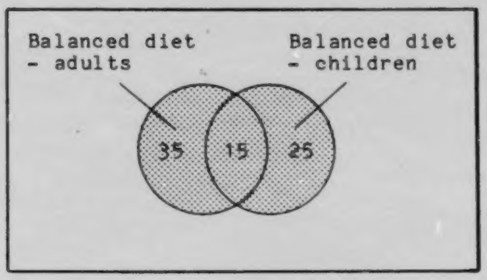
Keyword and Match Index Commands

The following examples show how the keyword and match index commands interact to retrieve selected portions of the itembank. (Other item characteristics such as the item type and format type can also be included in the retrieval process, but for the sake of simplicity only the keyword and match index parameters will be illustrated here.)

EXAMPLE 1 - Simple Keyword Selection

Keywords: 1 Balanced diet - adults
2 Balanced diet - children

Commands: Keywords 1 2
Match Index 1



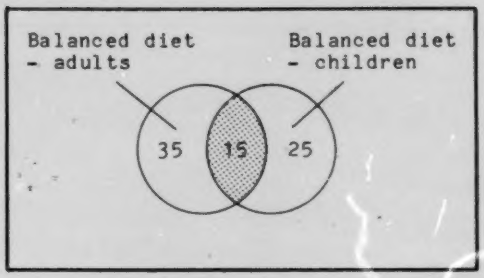
No. of items retrieved: 75 (35+15+25)

Explanation: The match index has been set to 1 which means that all items coded by either of the two keywords will be retrieved.

EXAMPLE 2 - Items Common to Two or More Categories

Keywords: 1 Balanced diet - adults
2 Balanced diet - children

Commands: Keywords 1 2
Match Index 2



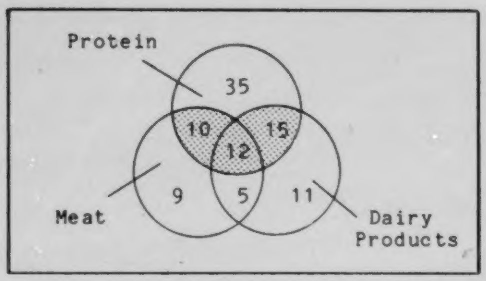
No. of items retrieved: 15

Explanation: A match index of '2' indicates that both of the specified keywords must be present in the code of any items which are retrieved.

EXAMPLE 3 - Specifying a Keyword of Primary Interest

Keywords: 14 Protein
7 Meat
8 Dairy products

Commands: Keywords +14 7 8
Match Index 2



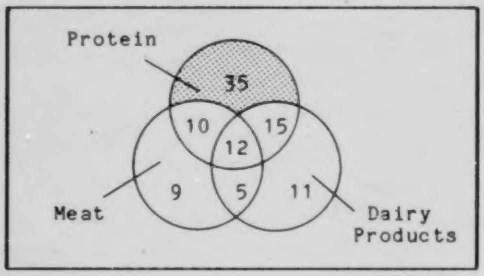
No. of items retrieved: 37 (10+12+15)

Explanation: The primary interest keyword (+14 - Protein) must be present in the code of any retrieved items. In addition one other keyword (7 - Meat or 8 - Diary Products) must be present.

EXAMPLE 4 - Specifying Excluded Keywords

Keywords: 14 Protein
7 Meat
8 Dairy products

Commands: Keywords 14 -7 -8
Match Index 1



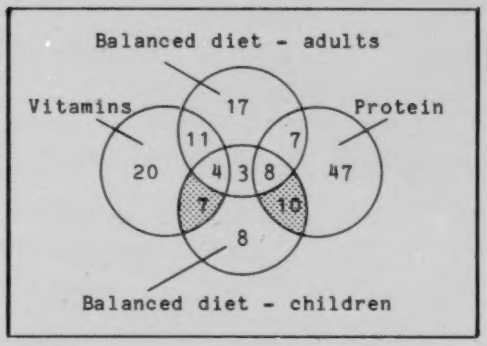
No. of items retrieved: 35

Explanation: Only those protein items which do not have a code for 'meat' (-7) or 'dairy products' (-8) are retrieved. This example and Example 3 can be seen to result in two independent groups of protein items. The match index is set to '1' because only one keyword (14 - Protein) is used to actually retrieve items.

EXAMPLE 5 - Both Primary Interest and Excluded Keywords

Keywords: 1 Balanced diet - adults
2 Balanced diet - children
14 Protein
15 Vitamins

Commands: Keywords -1 +2 14 15
Match Index 2



No. of items retrieved: 17 (7 + 10)

Explanation: This example is not as complicated as the diagram might seem to indicate. The request is for items coded 'balanced diet - children' which are also coded for 'protein' or 'vitamins', but excluding those items which contain a code for 'balanced diet - adults'. The match index is set to '2' and indicates that the primary interest keyword (+2) and one other keyword (14 or 15) must be present in the items retrieved.

Current Use of the System

The health science itembank now contains close to 6000 items and instructors from the three participating schools are continuing to contribute new items on a regular basis. There seems to be general agreement that the coding and retrieval scheme is adaptable to the instructional needs of each institution.

Over 150 tests are currently available on the system. There are two ways in which each test can be used. It can be given in printed form to a class of students or presented on a computer terminal for a student to take on an individualized basis. The individualized test makes it possible to provide the student with feedback after each response.

Additional itembanks are being developed for courses in business and the behavioral sciences. It is interesting to observe the reaction of instructors to the task of building an itembank. Frequently there is a tendency to view the task as one of simply duplicating the existing course structure in the coding of items. For example, a course may consist of units with an examination accompanying each of these units. The instructor may therefore feel the coding task is completed when all items used for a particular unit examination are given the code for that unit. Such a coding procedure, although quick and easy, defeats the real purpose of the coding and retrieval scheme described in this paper. It is the content of the item rather than its current use in a particular curriculum which should determine the codes applied to it. The task of determining these codes, although time consuming, is worth the effect. The result is a bank of test items which can be used in a wide variety of instructional situations.

AN INSTRUCTIONAL STRATEGY FOR COMPUTERS AND CALCULATORS WITH APPLICATIONS

Samuel W. Spero
Department of Computer and Information Science
Cleveland State University
Cleveland, Ohio

Abstract. An instructional strategy for using both computers and calculators together in teaching mathematics has been developed and tested. In this strategy the computer is used to generate problem sets for students and simultaneously to generate the solution sets for use by the teacher. The problem sets which use "real" numbers can be conveniently solved by the students using calculators and algorithms developed in class. Examples of the strategy in the areas of polynomial equations, simultaneous equations and matrices , and graphical addition are presented in detail.

Introduction

We have developed and tested an instructional strategy for use in the teaching of various topics of mathematics which employs both computers and calculators in a cost effective and pedagogically effective manner. While the strategy is used primarily for evaluation of students, it is a highly motivating strategy and can be put to good use for homework assignments as well. While we ourselves have used the strategy in mathematics courses at a local community college, other teachers with whom we have shared this strategy have used it in junior and senior high school mathematics courses. We are also examining its potential for university level courses. Topics to which we have applied this strategy include polynomial equations and elementary matrices in college algebra and trigonometry and graphing in the study of elementary functions. The compuerized segment of the strategy can be implemented on virtually any type computer, including the recently introduced microprocessors, for which there is an algebraic language like BASIC and whose hardware includes a printing terminal. We have used the BASIC language because of the convenience of the RND (random number generator) function which is crucial to the implementation of this strategy. The programming of the computer is straightforward and does not require files or even a great deal of internal memory. Furthermore, the strategy can be easily integrated into existing curricula and methods of instruction.

In this instructional strategy the computer is used to generate problem sets-a different problem set for each student. Simultaneously--and this is the unique feature of this approach--the computer generates the solution set. The teacher uses these individualized solution sets, which include the student's name, for evaluating the student's work.

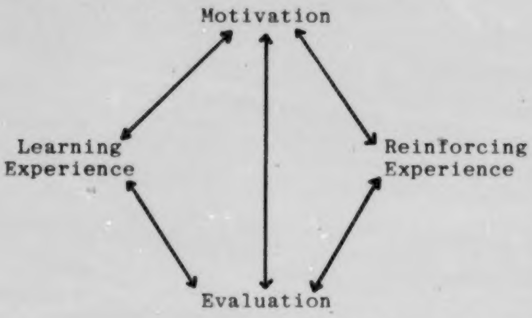
In order to provide the students with more realistic and more general problem sets, the computer generates problems that are more complicated computationally than those the student encountered in the textbook itself. With the help of the calculator and algorithms developed in class, even these more complicated problems can be solved quickly if the student understands the algorithm. For algebra a four-function calculator is sufficient; for elementary functions a scientific calculator is required.

Because each student's assignment is unique, we permit the students to work on these problems outside of class. This relieves a great deal of pressure that characterizes the in-class testing situation. This approach also saves a considerable amount of class time. This also permits a criterion-referenced approach to this material in that students may request additional problem sets with little loss of credit if they cannot complete their first problem set without help. Both the individualization and the criterion-referenced approach virtually eliminate students' interest in cheating.

In this paper we will provide an instructional rationale for this strategy and then describe some of the applications of this strategy which we have used. The interested reader may obtain copies of programs and/or the articles in which the application was described by contacting the author.

A Perspective on the Instructional Strategy

The instruction process includes four steps which must be present in all teaching if it is to be effective. The diagram below illustrates these four steps and their inter-relationships:



While in specific situations there may be variations on this theme, the four steps must be accounted for in all good teaching.

To implement these four steps in mathematics teaching we traditionally start with a teacher and a text. The teacher uses a blackboard, lecture, various discussion strategies, etc., to achieve motivation, to implement a learning experience or a reinforcing experience, and to perform evaluation. Homework, math labs, quizzes and tests, fields trips, films, all have their roles in helping implement one or more of the four steps of the instruction process.

The present instructional strategy has elements of all four steps. We have found that use of the calculator, especially the student's own calculator, is highly motivating. The student enjoys working with his or her calculator and even looks forward to assignments permitting them to learn to use their calculators more effectively. Studying various algorithms in mathematics problem solving not only helps the students learn to use their calculators more effectively, it also helps them learn the mathematics. While we generally provide a learning experience without using the calculator, the use of the calculator is certainly a reinforcing experience. Finally the primary use to which we have put this strategy is in the area of evaluation in which it has certainly played a significant role with respect to those objectives which can be evaluated in this way.

The Instructional Strategy with Applications

The instructional strategy which we have developed includes three phases. Phase I is the presentation of the problem to the students. The problem is generated by the computer with a different problem being generated for each student. As you will note the problems are personalized, i.e., each student's name is printed out with the problem. This is not just being cute; the name identifies the solution set for evaluation purposes.

Phase 2 is the solving of the problem by the students. The students apply problem-solving algorithms introduced in class in conjunction with their calculators to develop the solutions to the problem presented to them. These solutions are then submitted to the teacher for evaluation.

We have found it useful to only provide the students with a general outline of the algorithm. Each student is then required to work out the actual algorithm (in terms of button-pushing) for their own calculators. They turn these button-pushing flowcharts in to the teacher as part of their assignments. A great deal of learning takes place when the students must adapt the general algorithm to their own calculators. Because it is their own calculators, they are highly motivated. In fact in cases where students have owned programmable calculators the students on their own initiative have programmed the entire problem.

Phase 3 is the evaluation of the student's work by the teacher. Using the personalized solution set prepared by the computer at the same time as the problem was generated, the teacher is able to conveniently evaluate each student's work without having invested any more than the normal time that would have been allotted to this process.

These three phases of the instructional strategy are illustrated in the following three case studies which have been or are presently being used by the author in class.

Polynomial Equations - Case Study 1

Phase 1--Each student is presented with a third degree polynomial equation as in the example below. The student is told to find the first root of this equation by a half interval search technique. This root is then to be factored out from the equation by using synthetic division. The resulting quadratic equation is solved using the quadratic formula. The student is also instructed to find the extrema (both maximum and minimum) of the given equation using a formula which is given.

```
THE POLYNOMIAL FOR RICHARD BUSH IS

X 3 + -10.34 X 2 + 5.7008 X + -.3904
```

Phase 2--In class the students are taught a technique for rewriting the polynomial so that it can be evaluated conveniently for any value of the independent variable using the calculator. They also review synthetic division and the quadratic formula from the point of view of using their calculators with these procedures. The equation for the extrema in terms of the derivative of the function is given to them with some explanation but no details.

Phase 3--The figure below is a copy of the teacher's printout for the example shown in phase 1. Note the student's name, the student's problem, and the roots and the required extrema.

```
THE POLYNOMIAL FOR RICHARD BUSH IS

X 3 + -10.34 X 2 + 5.7008 X + -.3904

THE ROOTS ARE
.08                .5              9.76

THE EXTREMA ARE
FOR X =  .287672     Y = .417678
FOR X =  6,60566     Y = -125.68
```

Computer program--The computer program first generates the roots using the random number generator (the RND function in BASIC). For convenience the roots are restricted to values between ± 10. They are also chosen to be real. These restrictions can be easily removed or modified. With the roots selected, the coefficients of the polynomial can be calculated as can the extrema.

Simultaneous Equations and Matrices-Case Study 2

Phase 1--The problem presented to the students required that they find the solution set to a set of three simultaneous equations in three unknowns. This problem was presented to them in the context of a unit on elementary theory of matrices. They were therefore also required to determine the inverse of the matrix of coefficients and the determinant of the matrix of coefficients. Since these arise directly out of the solution technique the students were taught, it involved no substantial extra calculations. An example is provided below:

```
SIMULTANEOUS EQUATION PROBLEM

THE SYSTEM OF EQUATIONS FOR JOHN SMITH IS

76 X + 31 Y + 12 Z = -6547

62 X + -13 Y + -53 Z = 2465

3 X + -42 Y + 50 Z = -2524

DETERMINE THE SOLUTION SET FOR THE ABOVE
SYSTEM.  ALSO DETERMINE THE INVERSE OF
THE COEFFICIENT MATRIX AND ITS DETERMI-
NANT.
```

Phase 2 --

The students are taught to apply the Gauss-Jordan technique for solving a set of simultaneous equations. This is a matrix technique derived from the classical technique of Gaussian Elimination. Using elementary transformations on the original matrix the student arrives not only at the solution set but also at the inverse of the original matrix of coefficients. The determinant is calculated as the product of the diagonals at a certain stage of the process. The students learn to use their calculators not only to perform the essential arithmetic but also to implement the very crucial pivot operation which lies at the base of the Gaussian Elimination method.

Phase 3--The figure below is the output for the teacher:

```
THE SYSTEM OF EQUATIONS FOR JOHN SMITH IS

76 X + 31 Y + 12Z = -6547

62 X + -13 Y + -53 Z = 2465

3 X + -42 Y + 50 Z = -2524

DETERMINE THE SOLUTION SET FOR THE ABOVE
SYSTEM.  ALSO DETERMINE THE INVERSE OF
THE COEFFICIENT MATRIX AND ITS DETERMI-
NANT.

THE SOLUTION SET IS  -50   -53   -92

THE INVERSE OF THE COEFFICIENT MATRIX IS
8.20811E-3   5.86212E-3   4.2439E-3
9.3012E-3   -1.07425E-2  -1.36193E-2
7.32052E-3  -9.3754E-3    8.30515E-3

THE DETERMINANT OF THE COEFFICIENT MATRIX
IS -350385.
```

Computer program--Using the random number generator the computer program first generates the solution set then the matrix of coefficients. The values are restricted to integers between ± 100. The right hand side of the set of equations is then computed using the solution set together with the matrix of coefficients. The Gauss-Jordan method is then applied within the program to determine both the inverse of the matrix of coefficients and the determinant.

Graphical Addition--Case Study 3.

Phase 1--The topic of graphical addition is included in most textbooks of trigonometry. Traditionally it has been very difficult to do more than touch upon this subject and, perhaps, provide a demonstration or two. The reason is that the problems are so tedious to solve. Yet the topic has important applications in many areas of physics and deserves a more intensive look. Using calculators and the computer we were able to provide each student with his own problem to solve. The problem given each student was of the form:

$$Y = A\sin Bx + C\cos Dx$$

where A, B, C, and D were integers randomly generated by the computer. Bounds were also selected for the domain in which the student was required to plot the given function. Furthermore the student was told to plot only 20 points. An example is included below.

```
GRAPHICAL ADDITION

STUDENT -  SAM SPERO
GRAPH THE FOLLOWING FUNCTION BY GRAPHI-
CAL ADDITION OVER THE INDICATED DOMAIN.

Y = -6 SIN(8)X + 8 COS(7)X

THE DOMAIN OF YOUR PLOT SHOULD EXTEND
FROM X =-1.7 TO X=2.3

PLOT 20 EVENLY SPACED POINTS WITHIN
THE DOMAIN.
```

Phase 2--The primary computational activity in this exercise was the preparation of a table of values from which the graph of the given function could be prepared. The students were required to record not only the values of X and Y but also the values of the component functions. The student was required to plot these as well. The algorithm they learned to use with their calculators permitted them to perform this activity in the minimum number of steps.

Phase 3--In this case the computer printed the graph of the function within the given domain. This was the same graph the students were required to plot. As expected the computer-generated graph corresponded very closely to the graphs drawn by the students which made the exercise very easy to grade. Interestingly the students asked to keep the computer-generated plots after the exercise was completed. The example for the above problem is included below.

```
PROBLEM FOR STUDENT - SAM SPERO

Y = -6 SIN (8)X + 8 COS(7)X

THE DOMAIN OF YOUR PLOT SHOULD EXTEND FROM
X =-1.7 TO X=2.3

PLOT 20 EVENLY SPACED POINTS WITHIN THE
DOMAIN.

-14                                      14
--------------------------------------
                    I               *
         *          I
  *                 I
                    I     *
                    I                 *
                *   I
  *                 I
                    I*
                    I             *
                    I *
        *           I
               *    I
                    I      *
                    I   *
                *   I
                *   I
                   *I
                    I*
                    I   *
                    I*
             *      I
```

Computer program--The values of A, B,C, and D were generated using the random number generator. For convenience their values were restricted to lie between ± 9. The domain of the plot was specified by choosing randomly an upper bound somewhere between 0 and 7 (about 2 π) and then choosing randomly the distance between upper and lower bounds, also something between 0 and 7. The lower bound was then simply the upper bound minus the distance between them. The graph was drawn by first scaling the value of the function to the selected output line of the teletype which was chosen to be 60 spaces for aesthetic reasons. Once the value of the function was appropriately scaled it was only a matter of tabbing to the value of the function and then printing the appropriate character.

Summary and Conclusions

The instructional strategy we have presented in this article--which includes the use of the computer to generate problem sets for students and solution sets for teachers, as well as extensive use of calculators to solve the problem sets--

has had a number of ramifications beyond the original intent.

The original intent of this approach was to permit more effective evaluation of student learning. While we have not as yet tested this approach formally, we intuitively feel that the approach does provide more effective evaluation. Not only are we able to provide students with their own individual problem sets which they can solve at home, but also we have been better able to crystalize our own objectives for the students. By being able to focus directly on the learning objectives in instruction, the students are more aware of what is expected of them. This leads to improved learning.

In addition encouraging students to use their own calculators has increased student motivation considerably. Students spend time developing effective algorithms for solving problems with their own calculators. As a result they have learned more mathematics. Thus we have not only developed a tool for evaluation, but we have also been able to motivate our students and help them learn more. Considering this in the light of the model presented at the outset of this paper has led us to conclude that we have identified a useful strategy for mathematics instruction.

In using the computer to generate problem sets for students and solution sets for teachers, we feel that we have touched upon an approach which places the computer in a role for which it is unquestionably well-suited. In this role it provides the instructor with technological assistance in completing a task which he could not reasonably perform himself. To provide this assistance we require only a minimal computer facility which makes the computer's assistance cost effective. The instructor on the other hand reaps maximum benefit. Using the computer in this way we have been able to motivate the student; we have been able to help him learn concepts more effectively; and we have been able to evaluate the effectiveness of the entire process. What more can we ask of computer-assisted-instruction?

Footnotes

1. Samuel W. Spero, "Computers and Calculators: Polynomial Equations," Proceedings of the National Association of Users of Computer Assisted Learning (NAUCAL) Annual Meeting, Dearborn, Michigan, Nov. 2-5, 1977.

2. Samuel W. Spero and Mary Koehler, "Computer-Generated Problems Sets: Simultaneous Equations and Matrices," The Two-Year College Mathematics Journal, Vol. 8, No. 3, June, 1977.

3. Samuel W. Spero, "Computers and Calculators: A Strategy for Combined Usage in Trigonometry," Proceedings of the Eighth Conference on Computers in the Undergraduate Curricula, Michigan State University, Lansing, Michigan, June 20-22 1977.

4. Samuel W. Spero, "Calculators and Computers: Graphical Addition," submitted for publication in The Mathematics Teacher.

322

A VERSATILE BASIC PROGRAM FOR TEST GENERATION USING A SMALL TIME-SHARING MINICOMPUTER SYSTEM

James E. Van Verth
Canisius College

This paper describes a program for generation of a series of unique but equivalent tests, written in Data General Extended BASIC for use on a DG Nova 2 minicomputer with associated storage disk. Considerable variety, without the necessity for storing large numbers of questions is provided by: a) using questions with alternate endings, one of which is selected when the test is written; b) generation of numerical data from specifications included in the question text, and c) allowing questions which cover more than one category of test item, with questions selected to cover all designated categories with as little overlap as possible.

The design of computer programs which can assemble appropriate tests from a bank of questions has been a subject of interest in academic circles in recent years.[1] The writing of the program described here was inspired by the author's experience with the teaching method known as the Personalized System of Instruction (PSI). In this system, the course is divided into small units (usually 15-20 per semester), and students are tested individually on demand to assure mastery of each unit. Since retests are provided without penalty until mastery is achieved, as many as 120 or more tests must be written for a two-semester course. The writing of these tests is one of the more time-consuming and demanding aspects of the operation of a PSI course, particularly if the instructor wishes to provide a new set of tests each time the course is taught.

Although the present program was designed with a PSI course in mind, it could be used with any type of course. It is necessary to divide the course into sections or units, each unit comprising an amount of material appropriate for testing. Test questions are assigned to categories according to the specific aspect of the subject matter which is being tested. In a PSI course, the categories correspond to the performance objectives which are customarily specified for each unit. In the test generation program, questions are referenced by category number, and by question number within a category. When a test is generated, questions are selected so as to cover all desirable categories.

Since the questions that we teachers write are often designed to test more than one specific objective (in fact, some of the best questions are of this type), it was clear that a good deal of flexibility could be obtained if this type of question could be handled by the program. This was done by allowing the question writer to assign several applicable categories (up to three in the present version) for each question, which are stored with the question text. When the test is assembled, the categories are examined, and questions are selected so as to avoid duplication of any categories, unless such duplications are unavoidable owing to limitations in the variety of questions on file.

A common solution to repeated testing of the same item is to write several different versions of what is essentially the same question; e.g., different information may be provided in each version, or a different question based on the same information may be asked, etc. In the present program, this is handled by allowing several alternate endings to be provided for a given question. As a simple example, a question could begin, "What is the capital of each of the following states?" This would be followed by several lists of states. One of these lists would be selected when the question is printed. An "ending" can be as small or as large a portion of the question as desired; in fact, several versions of the whole question could be filed under a single reference number in this way.

In questions involving calculations, additional variety is provided by having the computer generate the numerical data, if desired. When the question is filed, information on upper and lower limits and number of significant digits is given for each number to be generated. For convenience, this information is included in the question text at the location where the number is to be inserted.

The program is written in Data General Extended BASIC for use on a Nova 2 minicomputer, operating under Data General's Real Time Disk Operating System. By means of a separate filing program, questions are stored on a moving head cartridge disk having a capacity of five megabytes of fixed storage. An additional five megabytes of removable storage and a floppy disk reader are available. About 7300 bytes of core space is available for each of the time-sharing terminals.

Three randomly accessible disk files are used. One each is used for the storage of questions and the corresponding answers. The third is used as a directory; in it is stored a matrix for each unit (the "directory matrix") containing the numbers of the initial file records for the questions and answers, referenced by category number and question number.[2] The same initial record number is used for both the question and its corresponding answer.

While the randomly accessible files enable direct access of individual questions, they have the disadvantage that the record size must be fixed at one initially specified value. If the size is set large enough to hold any possible question, a great deal of space is wasted in the storage of shorter questions. The best solution seems to be to set the record size at a value suitable for most anticipated questions, and to store larger questions in more than one record. A pointer to the next continuing record is appended at the end of each record prior to the final one.

OPERATION OF THE PROGRAM

Preliminary Procedures

After input of the course number, unit number, number of tests to be generated, number of copies of each test, and number of answer keys, the appropriate files are opened and the directory matrix is read into the core. The matrix is searched by category and a list of categories for which questions have been filed (record number > 0) is made.

Test Assembly

For each test to be assembled, the following basic procedure is followed: The list of file categories is first scrambled randomly.[3] Then for each category, in the scrambled order, a question number is randomly selected from among those for which questions are stored. Then the record number of that question is obtained from the directory matrix and placed in another matrix (the "assembly matrix") in order of selection.

Checking for Category Duplication

The specified categories for each question chosen are also placed in the assembly matrix along with the record number. Each question has been automatically filed under the highest numbered of its specified categories for which there is room in the directory matrix. Before a question is chosen from a given file category, that file category is compared with all the categories of previously selected questions. If a duplicate is found, that file category is skipped. If none is found, questions are chosen randomly from that file category until one is found whose other categories do not duplicate any of the categories of previous questions. In order to avoid entrapment in a continuous loop, if no such question is found in a given number of attempts, the questions in the file category are examined systematically. If none with unique categories is then found, the duplication is accepted, and the last randomly chosen question is used.

History of Use

If several tests are to be generated on the same unit, it is usually desirable to avoid repeating the same question on more than one test. To provide a record of use, a history matrix is constructed with a row for each file category, and enough columns to hold the question numbers used on successive tests. In the question selection process, the history of each file category is examined, and questions which have been used before are rejected. The history matrix in the present program has five columns, permitting the construction of up to six tests containing no duplicate questions. If less than six questions are available in a given catefory, duplications are permitted after all questions have been used once. If alternate endings for a given question are available, a different ending will be used in each repeat of the question (see below).

Printing of Tests

After assembling the test, the file record of each question in turn is extracted from the assembly matrix, and the question text is read from the file as a string variable. Since the BASIC system does not permit printing of strings which exceed the page size limitation of 132 characters, the text string is printed in sections, each consisting of one line of text. The end of a line is marked by a carriage return which is included automatically when the question is filed. The same process is followed for other copies of the test and for printing of the answer key.

Alternate Endings

When the question is filed, alternate endings are denoted by typing a right parenthesis as the first character of the first line of each ending. Separate answers for each ending are indicated in the same way in filing the answer.

As the test is printed, the first character of each line is examined. If a right parenthesis is found, the remainder of the text is scanned and the number of right parentheses following carriage returns is counted. One of these is selected randomly, the beginning of that variation is found, and the printing of the text is continued at that point until either the next initial right parenthesis or the end of the text is encountered. The number of the ending used is remembered by dividing it by 10 and appending it as a decimal increment to the question number stored in the history matrix. It can then be retrieved for printing of subsequent copies and of the answer key. The history is consulted each time an ending variation is chosen, to assure that, if the same question has been used before, a different variation is selected each time, until all endings have been used.

Number Generation

For numerical data to be generated at the time of printing, the limits and number of digits are given within brackets, separated by slash marks, at the appropriate location in the text. When a left bracket is discovered within a line to be printed, transfer is made to a subroutine which inspects the bracketed region and extracts the information therein. (If the bracketed material does not conform to the proper format, an ON ERR statement is used to transfer control back to the regular printing routine, and the bracketed portion is printed as filed.) A random number generated within the given limits is reduced to the specified number of significant digits by the following manipulation: The number is multiplied by 10^{12} in order to insure its expression in exponential format, and then converted to a string variable. The portion of the string before the letter "E" is truncated to the required number of digits, and then recombined with the exponential portion reduced by twelve powers of ten. The resulting number is printed, and also written to a disk file for use in subsequent copies of the test and for answer calculation.

Before printing the text of an answer, the number file is read for any numbers which were generated during the printing of the corresponding question. Each number is assigned to a variable, N1, N2, etc., which can be used in the calculation of the answer(s). In the answer text, the argument of an equation for calculation of the numerical answer using the proper variable designations is included between brackets at the proper location. When the bracketed portion is encountered in printing, it is converted to a program statement by preceding it with an appropriate statement number and a variable assignment ("A="). After putting the statement on a disk file in ASCII format, it is entered into the program and immediately executed for calculation of the answer, which is then printed on the answer sheet.

---

[1] Gerald Lippey, Ed., "Computer-Assisted Test Construction," Educational Technology Publications, Englewood Cliffs, N.J., 1974.

[2] This method of referencing the text files was suggested by Mr. Michael Szymendera.

[3] Using the efficient "shuffling" algorithm of Moses and Oakford, described by Donald E. Knuth in "The Art of Computer Programming," Vol. 2, Addison-Wesley, Reading, Mass., 1969, p. 125.

AUTOMATING EXAMS FOR A STATISTICS COURSE
THE BEGINNING OF A CASE STUDY

J.A. Warren, R.D. Michener, and A. Bugbee
Office of Academic Computing
University of New Hampshire

An account is given of the general kinds of things that were anticipated before automating exams for a statistics course. A companion paper relates what actually happened.

This paper is the first of a two part account of a case study in automating exams for a course. In this more general paper we describe the steps involved in this process and the efforts that we anticipated for our case study. In the second paper we relate what actually happened when we automated a brief statistics course based on the book How to Lie with Statistics (Huff, 1954).

The case discussed is a statistics course. The process described will be aided by question and answer files and programs belonging to the Statistics Test Item Collection System (STICS*). But, much of the process described applies to other disciplines. The contents of a question-answer file will vary from discipline to discipline, but the nature of the process and the programs needed shouldn't vary from discipline to discipline.

Automating Exams

We plan to produce exams that consistently reflect the objectives for a course in a manner that does not involve the active intervention of the person(s) currently teaching that course. We plan to provide answer guidelines for each exam, but do not propose to provide computer grading or computer record keeping. We plan to provide the option of having exams delivered at a computer terminal, but do not regard doing so as crucial. In fact, we will describe a process that can be implemented without the direct involvement of a computer.

Purpose

Our principal motive in automating exams is that we want consistency in what is required of students in a course. We do not want course standards to vary wildly from instructor to instructor or for that matter with changes in the same instructor. The standards a student must meet should not be dominated by the particular person who happens to be teaching that student. We want the people responsible for a course to agree on a minimum set of requirements and a mechanism that will provide exams meeting those requirements regardless of changes in instructors.

THE PROCESS

To illustrate the process of automating exams, we have made up a course that is new to ourselves. Because non-statisticians as well as statisticians seem to find it readable and amusing, we decided to use How to Lie with Statistics as a starting point for our course. We will develop our illustration in the context of an imaginary course that attempts to promote student adoption of the critical, questioning outlook expressed in How to Lie with Statistics.

In formulating our course we would like to avoid as much unnecessary detail as possible. However, some details may be helpful so that what we do later won't seem completely arbitrary. We are thinking of:

---

*STICS is a project partially supported by NSF-Diagnostic and Instructional Services for Undergraduate Students in Statistics (SED 76-12191)

- a total enrollment of 100 students or less per term
- an institutional setting in which it is reasonable to use essay questions and to assume access to at least pocket calculators during exams
- a testing scheme that gives students access to potential exam questions prior to exams and permits repetition of student exams

The process that we are illustrating doesn't require these conditions, but they are conditions that will influence our illustration. We will leave unspecified many other conditions such as time allocated to this course, number of credits, etc.

Step 1 — Setting Objectives. We will reread How to Lie with Statistics and use it to form an initial set of objectives according to the guidelines provided by Mager (1962). We will try to specify what we want students to do when they finish our course. Eventually, we expect to arrive at statements of 5-20 objectives as a trial set that almost certainly will be modified later. We will also classify these objectives according to approximate relative importance.

Step 2 — Selection of a Question Frame or Collection to Reflect Our Objectives. STICS now can provide:

- a National Statistical File that consists of around 2000 questions and answers that have been classified according to statistical topic and question characteristics such as question type, difficulty, etc. (eventually this file should have over 6000 items)
- SELECT, a program that can be used to extract questions and answers from the National File to form a much smaller local file or sampling frame consisting of just those items that meet our requirements

In an ideal world, we would immediately choose keyterms that stated exactly the statistical topics matching the objectives of our course. Then we would add further specifications about question type, degree of difficulty, math prerequisites, etc., and pop out a collection of items that were fully consistent with our objectives. In fact, we expect to have some of the same kinds of adventures that we have with the Yellow Pages in a telephone directory or a card catalog in a library. We will try a number of keyterms until we find some that seem to approximately agree with our objectives. We will use these keyterms to extract more questions than we expect to need. We will expect to discard many of the items that come through this rough filter. Depending on the number of items that qualify according to keyterms, we may or may not use question characteristics at this stage of selection. Our hope is that we can use SELECT to arrive at a collection of 100-400 items that will include most of the items that will be needed for preparing exams. This collection of items is a first approximation of the sampling frame that will be used to prepare exams for our course. We will call this collection FRAME(1).

This step of extracting, say, 200 questions from a collection of 2000 or more supplies at least one frustration and at least one dilema. The frustration is that there is no easy way to be sure that this transition will be accompanied by the sound of trumpets. It should. If the extraction process is successful it means that one to two hours use of an interactive program has been successfully traded for the man weeks that Hammer and Henderson (1972) estimate are needed for developing an adequate test bank. The dilema is that even though use of SELECT has the potential for sizeable savings in effort, it also presents problems on how long an interactive user should be expected to sit still. Suppose that a user has entered specifications that are met by, say, 30 items. If the user immediately prints all 30 items it will be possible to immediately decide if those items are acceptable. But, the average length of an item is around 18 lines (72 characters per line). So it will cost around 20 minutes on a 30 cps terminal to look at all those items. If these items aren't printed, chances are increased that unacceptable items will survive this part of the extraction process. Our current solution is to allow easy printing of abstracts (average length two lines) with the expectation that the price paid for limiting the time needed to run SELECT is that some unacceptable items will pass this stage of the selection process.

Step 3 — Editing the Question Frame. Following the first execution of SELECT, we will print the full set of questions and answers making up FRAME(1). When FRAME(1) is carefully inspected, we expect to find:

- clearly inappropriate items that should be deleted
- nearly appropriate items that will be acceptable following minor changes
- potentially appropriate items that require appreciable effort to become useable
- items that seem quite reasonable for this course, but do not match our original set of objectives. These items will stimulate a reexamination and possibly changes in our original set of objectives.

Following this inspection, we will probably make deletions and minor changes on FRAME(1) using conversational editing programs included in STICS. We then will have FRAME(2) which will be, say, 1/2 to 3/4 the size of FRAME(1).

Step 4 — Setting Up for Exams, More Editing. There are two major notions that bring about the transformation of FRAME(2) into FRAME(3). We will illustrate these notions using a hypothetical situation chosen for convenience.

1. Only two kinds of keyterms are needed for purposes of generating exams and neither type will be found in the National File. For generating exams it is convenient to be able to identify each item in FRAME(2) according to objective and question group. Each question can be thought of as belonging to a cell in a table or matrix having objectives as rows and question groups as columns.

A question group generally will consist of items thought to be similar in terms of difficulty or time required. e.g., An exam might be made up of three question groups:

Group 1:	items thought to require five minutes or less
Group 2:	items thought to require ten minutes and emphasizing comprehension
Group 3:	items thought to require ten minutes and emphasizing computational skill

Group 1 might include true/false, multiple choice, and other short question types. An exam might be made up using five items from Group 1, one from Group 2, and one from Group 3.

When all items have been classif[illegible] in terms of objectives and question groups an inventory can be produced such as the hypothetical data in Table 1.

TABLE 1 — HYPOTHETICAL INVENTORY OF ITEMS IN FRAME(1) CLASSIFIED ACCORDING TO OBJECTIVE AND QUESTION GROUP. (TABLE ENTRIES ARE NUMBERS OF ITEMS.)

Objective	Question Group 1	2	3	Number of Items Dealing with Objective
1	50	5	20	75
2	20	7	20	47
3	15	4	1	20
4	15	0	6	21
No. of Items in Question Group:	100	16	47	
OVERALL TOTAL 163				

Notice that this inventory indicates that unless FRAME(2) is further modified there is a very unequal emphasis on objectives. For every question related to objectives 3 and 4, there are two questions dealing with objective 2 and four questions dealing with objective 1.

2. The number of questions in a frame that address an objective should be roughly proportional to the importance of an objective. While this condition isn't essential it is helpful especially in cases when students are given the complete question frame prior to exams. e.g., Suppose that objective 4 in Table 1 is thought to be equal in importance to objective 1. If a student expends the effort needed to understand all the questions in that frame, we would like about the same effort to be directed to each objective. Having four times as many questions for objective 1 as for objective 4 does not seem a promising way to achieve this.

Suppose that the relative importance weights attached to objectives were:

WEIGHT
Objective(1) = 3, most important
Objective(2) = 3, most important
Objective(3) = 2,
Objective(4) = 1, least important
TOTAL WEIGHT = 9

If numbers of questions associated with a question are proportional to an objective's contribution to total weights, then around 1/3 of the questions should be concerned with objective 1 and 1/9 concerned with objective 4.

In order to bring about proportionality between number of questions and the importance attached to various objectives the inventory in Table 1 needs to be adjusted. One way of doing this is to make additions and deletions so that proportionality is achieved within each question group. e.g., For Question Group 1 one of several acceptable sets of changes would be:

-Drop 17 items related to Objective(1) → 33
-Add 13 items related to Objective(2) → 33
-Add 7 items related to Objective(3) → 22
-Drop 4 items related to Objective(4) → 11

The numbers of questions related to various objectives might be adjusted similarly in Questions Groups 2 and 3. The temptation probably would be strong to neglect adding new items related to objective(4) since its relative importance is low.

In general, the transition from FRAME(2) to FRAME(3) will almost always require the attachment of keyterms for question group and objective. There seems to be little likelihood that there will be 1:1 correspondences between these keyterms and the statistical topic and question characteristic keyterms used in the National File. More often than not one objective will involve two or three topics and one question group will involve more than one question type.

The process of adjusting numbers of questions to the importance of objectives may not be generally needed. In cases where the question frame is given to students prior to exams we recommend that this step be taken. Further, we guess that the total number of questions distributed for an exam covering four to six weeks work should be on the order of 100 to 200 items. If too few questions are distributed, preparation for an exam can be reduced to a trivial exercise in memorization. If too many questions are distributed, use of the questions in studying for an exam will be discouraged.

Our expectations concerning the How to Lie with Statistics course are that the transition from FRAME(2) to FRAME(3) will involve five to twenty objectives and four to five question groups. We will not be surprised if this part of the process requires more than one iteration.

Step 5 — Generating Exams. In general, we want a program for generating exams that will randomly sample various strata, such as question groups, to prepare exams that will be at least roughly comparable in difficulty. If the collection of questions sampled has had numbers of questions adjusted according to the importance of objectives within each stratum, it is easy to arrange for the occurrence of questions to reflect the importance of objectives.

In automating exams for our How to Lie with Statistics course we will use one of the STICS programs (BEGET). We will use this program to generate, say, five distinct exams and prepare answers. Then we will ask if this is what we want students to face when they take exams for our course.

We will not be astounded if the answer is no. If so, we may find it necessary to return to any of the previous steps — including the setting of objectives and assignments of relative importance. While we do not expect to celebrate the necessity for revision, we do hope to be able to take comfort in having substantially completed the start-up portion of the process of automating exams. We expect that the contents of FRAME(3) should be at least 80 to 90 percent of what will be needed for FRAME(4) or later FRAMEs.

DISCUSSION

The process that we anticipate is flowcharted in Figure 1. Table 2 notes time required for a process, frequency of execution of a process, and crude indications of computer requirements. These materials and the preceding text are apt to mislead the reader in at least one important matter. Both allocate much, much more space to the start-up activities than to the pay off of generating exams over many years. We expect to invest a substantial amount of effort into these start-up activities, but we expect to have that effort repaid over several years in terms of effort saved in producing exams and improved consistency of standards.

TABLE 2 — ESTIMATES OF TIME AND RESOURCES NEEDED FOR AUTOMATING EXAMS FOR A COURSE CENTERED ON HOW TO LIE WITH STATISTICS.

Process	Guessed Time	Computer Resources: File Entries	Computer Resources: Memory and CPU Time	Number of Times Process Repeated	Relevant STICS Program
1. Setting Objectives	3-5 days	—	—	Few	—
2. Selection of Initial Frame	1-2 hrs.	2000 to 6000	Large[*]	Few	SELECT
3. Early Editing of Frame	1-3 days	100 to 400	Small	Few	TXTEDT
4. Setting up for Exams	2-3 days	100 to 200	Small	Few	KEYADD NEWQUE
5. Generating Entries	1-2 hrs per set of 5	100 to 200	Small	Many	BEGET

[*] Average core 80k bytes, 1 second CPU/10 minutes connect time on DEC 1090.

Our description of what we anticipate has involved use of the National File and programs of STICS. Computer requirements are noted in Table 2 as a reminder that most of this process need not involve use of STICS. The key part of the process that is aided by STICS is the formation of a first working frame from the National File. Once this has been done, the process is not a matter of using a file of thousands of entries. It is a matter of dealing with a file of at worst a few hundred entries. Once a frame has been edited and set up for generating exams, it usually will be a collection of less than 200 items that require just two identifiers, objective number and question group. Because of this, we hope that the role of STICS ordinarily will be to help people through the start-up steps to produce one or more machine readable files of the form of FRAME(3). We hope to see these files transferred to a wide variety of computers having their own exam generating programs (e.g., Eisenberg, 1977). Our current version of STICS is written in FORTRAN and runs on DEC10 systems. We hope to see BEGET equivalents running on many other systems including micro systems.

Finally, we admit having tried to simplify this description of the process of automating exams. We have tried to resist pointing out embellishments and alternatives in hopes of improving clarity. We do not pretend to have described an optimal procedure. We suspect that we will have to admit some of these simplifications in the following paper that relates what happened when we attempted to implement our proposals.

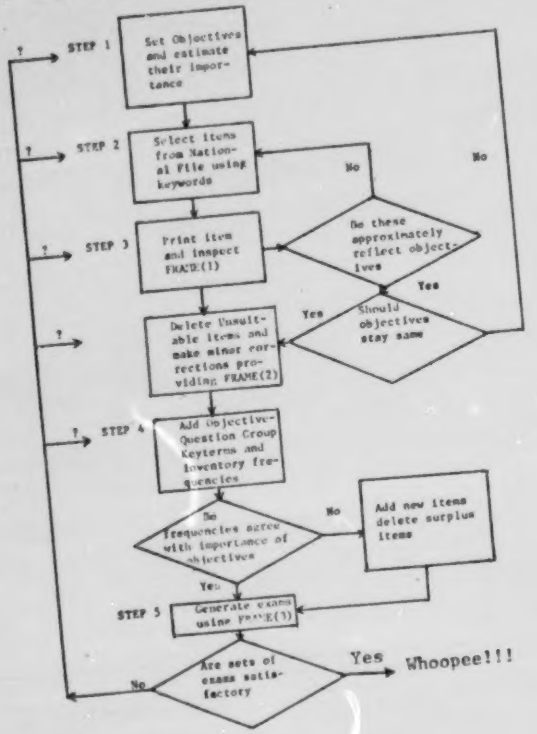

?
STEP 1
Set Objectives and estimate their importance
?
STEP 2
Select items from National File using keywords
No
No
?
STEP 3
Print item and inspect FRAME(1)
Do these approximately reflect objectives
Yes
?
Yes
Should objectives stay same
Delete Unsuitable items and make minor corrections providing FRAME(2)
?
STEP 4
Add Objective-Question Group Keyterms and inventory frequencies
Do frequencies agree with importance of objectives
No
Add new items delete surplus items
Yes
STEP 5
Generate exams using FRAME(3)
Are sets of exams satisfactory
Yes
Whoopee!!!
No

Figure 1

References

Eisenberg, B. (1977), "Final Exams — Let the Computer Write Them", Creative Computing, 3, No. 6, 103-107.

Hammer, M., and Henderson, C.O. (1972), "Beat the 'Load-Cram' Cycle with Computer Tests", College Management, October, 25-77.

Huff, D. (1954), How to Lie with Statistics, New York: W.W. Norton & Company, Inc.

Mager, R.F. (1962), Preparing Instructional Objectives, Belmont: Fearon Publishers.

COMPUTER ASSISTED TEST CONSTRUCTION : TWO YEARS OF USER EXPERIENCE WITH THE SOCRATES SYSTEM

W.Van Willis. California State University,
Fullerton CA 92634
Oliver Seely, Jr.,California State
University, Dominguez Hills, Ca 90747

SOCRATES computer assisted test construction has been available to the faculty of our nineteen campus system since 1975. Tests can be ordered by telephone and delivered by courier or generated batch mode from terminals at any campus. The test item banks have grown to more than 80,000 questions in 38 collections at this time. Users have found that the clerical burden of test production is lifted entirely and that the computational burden of scoring, item analysis, and item statistics maintenance is greatly diminished. Since the fall of 1976, no developmental support has been available for SOCRATES. Still, the number of frequent users has doubled, and 500 items per month have been added to new or existing banks. This suggests to us that self support for SOCRATES through user activity and input is possible.

SOCRATES[1] computer assisted test construction has been available to the faculty of the California State University and Colleges since the fall of 1975. Tests and other exercises can be generated either by telephone or by batch mode entry to the CDC 3300 central processor in Los Angeles from any of our 19 campuses. The resulting test may be directed to the user's campus to be produced on the local high speed printer, or it may be printed in Los Angeles on continuous form duplicating masters or standard forms and forwarded to the user.Tests requested before 8:00 PM are routinely delivered by courier the following morning. The test item banks have grown from 11 collections in 11 disciplines to 41 collections in 26 disciplines in less than four years. There are over 80,000 individual test questions. Approximately 500 new items are added to these each month. SOCRATES was designed to give the teacher a variety of pedagogical tools. Length: uo to 150 items may be requested in one test; content: items may be requested specifically, by subject matter block, by cross reference, or by special code; achievement: items may be selected by difficulty or behavior level;approach: true-false, multiple choice, interpretative, or problematical items are available. Items may be deleted and replaced with ease (but not reworded);parallel tests and multiple(scrambled) versions of the same test may be generated. Finally,most test items can be scored automatically, giving the teacher raw scores with ranks and two different levels of item analysis. Users employ all of these features in choosing and scoring their tests. The ability to conveniently edit and to block out exams by subject matter have proven to be extremely popular features.

It is interesting to note the acceptance rate by teachers in the various disciplines of the CATC approach used by SOCRATES. Somewhat surprisingly, those disciplines which historically have had the greatest familiarity with computer useage have been the slowest to accept CATC,and SOCRATES in particular: chemistry, mathematics,and physics. The greatest general acceptance and interest has come from business, accounting, and data processing. Other sciences, engineering, and the liberal arts fall in between. No simple explanation seems to be forthcoming for this phenomenon, since it involves teaching approaches, cousre lelev and content, and educational objectives. Suffice it to say that within every discipline we have found a number on enthusiastic supporters who have shown that the SOCRATES approach "works" for them, and that most of the negative bias which has surfaced revolves around a change in teaching habits,rather than pedagogical or other deficiencies in the CATC approach.

One question which arises from time to time is that of examination security. Our only reported problems have occurred between the local campus computer center which receives the completed exams and the final destination. No real breach of security has been found within SOCRATES, but persistent persons could of course break in, perhaps without activating our check system or the teacher. For most situations, the intruder would have to seek a specific exam,since the item banks themselves are so large that many possible questions on a given topic may exist. Thus for now, most of the security problems are local, and external to SOCRATES.

As regards the item banks themselves, several features have proven to be quite popular across all disciplines, and will be mentioned here. Questions which are keyed to specific texts enjoy a high use rate, perhaps sincetheir use reduces the time required by the teacher in screening potentially relevant items.The special characteristics which allow items to be selected via cross references or from special categories(such as a teacher's personally wrtiien items) are also mentioned by many users as good features. Some disciplines, such as accounting and history, write long questions and these can be reproduced, error free, by using SOCRATES as a tireless typist. Many questions are in a multiple choice format which lends itself to machine grading, and many teachers appreciate the time savings in grading as well as in using ready-made questions; in fact, some teachers have for many years expressed a desire to use such materials but could not afford the time necessary to construct the items or write the necessary software. Other teachers enjoy the use of the distractors (wrong answers, etc) as means of evaluating conceptual and skill problems within the class.

---

[1] Student Oriented Classroom Analysis and Test Evaluation System

The test request/editing process has been designed to minimize direct contact between user and computer. By making the computer transparent, we hoped to minimize effort and maximize results for the users who want an immediate service, not an apprenticeship in computer useage, with all the concomitant frustrations. It takes roughly one hour to learn how to request and edit SOCRATES exams, without any knowledge of computer languages or protocol.

Currently we are faced with increasing demand for interactive test generation and editing capabilities, and we are exploring several possible alternatives. With 19 campuses, this requires a dedication of many more resources than the present system. We are hopeful that the advent of low cost powerful minicomputers will help resolve the problem.

Our problem of exam scoring, which arises out of the multicampus nature of our system, has remained substantially the same since our inception. The variety of computers in the network, and the lack (until recently) of commercially available inexpensive computer compatible mark sense readers has forced us to ask most users to send exams off campus to be read onto magnetic tape for SOCRATES scoring. Since many teachers want instant results and are loath to release ungraded papers from their immediate possession, they grade their exams locally. This deprives SOCRATES of very usefulstatistical data, and deprives the teacher of an in-depth statistical analysis of student performance. Most campuses now are purchasing mark sense readers which are capable of being interfaced with a computer. It should be pointed out in passing that many teachers do not want a great detail of statistical information and analysis: therefore we give the option of two levels of analysis to accommodate those teachers who want a simple mean/variance analysis as well as those who want a student by student, item by item discriminative analysis.

Some disciplines, especially the sciences, experience quite severe problems due to the limited number of symbols available on the standars business print chain normally in place on line printers. Many of the problems have been solved with remarkable ingenuity by the chemists, physicists, and the like. We do employ a special print chain at the Los Angeles center for producing SOCRATES tests.This chain has a number of special sharacters which are useful, mainly to the sciences and engineering.

As faculty become used to applying the SOCRATES system to their classroom needs, we find their interest in developing materials for the item banks increases. This generally requires some training in order to correctly specify the item characteristics and punch cards in the correct format. We try to assist these people wherever possible, and refer them to a faculty member in charge of the relevant bank(the bank coordinator) for help. Once items are loaded into the system, they become accessible to all users, with the exception of special testing banks used only for competency exams. In this way, many new banks have been developed by interested faculty without direct financial support from SOCRATES. In addition, several publishers have made arrangements to have items keyed to their texts placed into the banks, mainly as a selling point. Interestingly, their position insofar as copyrighted material is concerned has been that because many of the questions cover material that is generally taught, therefore generally examined, there is minimal concern about copyright infringement. Indeed, certain questions can only be asked in a limited variety of ways, and the same problem has potentially existed between textbooks for many years. By the same token, we consider the bank contents by and large to be in the public domain and provide copies of item bank materials for useage outside our system.

The cost of generating a 100-item SOCRATES exam and scoring that exam for a 100 student class is currently approximately $12.00. This assumes that the exam is produced directly on spirit duplicating masters, and does not include paper costs. Each additional version of the same exam would cost another $1.00 to produce. The cost of placing items into a bank varies greatly depending on the kind of bank being built. General useage banks with extensive cross references and keywords usually cost from $6.00 to $12.00 for each item that is taken from a printed page through all the steps and placed into the bank.Textbook specific banks cost much less to produce, about $1.25 per item, since the only classification necessary is that of collection (text) number and chapter.

We have observed that the success of SOCRATES depends at least as much on a service provided by real people sympathetic to the needs of teachers as it does on a set of operational computer programs. Teachers are generally unwilling to become well acquainted with the idiosyncracies of computer operations and data processing center protocol, even if there is a promise of less work or more importantly,a promise of removing pedagogical barriers to innovation in teaching. It is clear that if they are not troubled by any aspect of the computer during their test request cycle, they make frequent and routine use of it. A good example of this occurred in the Marketing Department at one of our campuses. The secretary in that department was trained to request SOCRATES tests from a textbook-specific bank. We discovered

332

through an informal survey that the faculty who use that textbook get their tests exclusively from SOCRATES. The secretary will not even consider typing tests anymore; the faculty are happy to use the tests she provides them. The same textbook has been adopted by faculty at other campuses but the item bank is not used there, even though several announcements of its availability have been circulated.

Finally, we have observed that faculty want to receive services in addition to test retrieval. Some faculty want information on building their own banks; some wish to receive special printouts of banks by cross reference; others request immediate reprints of tests at local campus computer centers, after making the phone request to Los Angeles only minutes earlier. Still other people outside the system want copies of the item bank contents or magnetic tapes of the same.

There has been no support for software development since the fall of 1976. Until that time our funding was provided by the Office of New Program Development within the Chancellor's Office of the California State University and Colleges. We have entered an indefinite period during which statewide use will be observed before additional funding will be considered. Our support is now reduced to a document depository on one campus, a part time programmer at the State University Data Center who loads new bank items, the people who process SOCRATES test requests at the data center (only a small fraction of their time is devoted directly to SOCRATES), and various interested faculty who conduct an occasional workshop for users when requested. Despite the seemingly low level of support, and the problems described above, the numbers of frequent users has doubled, and items are added to the banks at the rate of about 500 per month. This suggests to us that self-support for development through user activity is possible.

COMPUTER-DESIGNED INSTRUCTION

THE USE OF COMPUTER-ASSISTED LESSONS FOR CALCULUS

Kendell Hyde
Weber State College

We have prepared a series of computer programs with accompanying lessons for students who have not had experience in computer programming. Each lesson is self-explanatory and can serve a variety of purposes including use as supplementary material in calculus courses. A master copy of each lesson is available so that an instructor may select a particular program and have this material reproduced for use by his class. The lessons are also used in a one-hour calculus laboratory. The laboratory allows students from any of the calculus courses to do additional work with the computer. The lessons are written so that the student can do much of the work on his own, thus allowing flexibility in scheduling and reducing the man-hours required to administer the program.

INTRODUCTION

The Department of Mathematics at Weber State College offers a variety of activities which require the use of calculators and computers. Students use CRT and hard-copy computer terminals or programmable calculators to solve problems related to the mathematics course in which they are enrolled. They may be required to write programs or use canned programs depending on the course and programming experience of the student.

The purpose of this paper is to explain the use of lessons designed for calculus students who have had no experience in computer programming. Students who use the programs are required to supply information and interact with the computer in order to obtain results. The use of the computer relieves the student of tedious calculations, reinforces mathematical concepts by providing numerical results and demonstrates the accuracy of numerical methods. We do not replace thoughtful reasoning by button pushing. Unlike programs which give a complete solution to a problem with a minimum of input, these programs require more details from the user and solve only the portion of the problem which requires tedious calculations.

DESCRIPTION OF LESSONS

Following is a brief description of the lessons that accompany the first quarter of the calculus course. Each lesson consists of instructions for running the programs, a set of exercises, a sample run of each program and a supplement for the instructor containing more detailed information about the programs and answers to the exercises. The instructions for running each program are listed on one page that can be duplicated and given to the students. The following is information given to the student for use of the program "NEWTON."

NEWTON

This program uses the formula
$x_{n+1} = x_n - f(x_n)/f'(x_n)$
to generate a sequence which in many cases will converge to a root of $f(x) = 0$. A table is printed showing the values of x_i and corresponding values of the function. From the table you can determine whether $f(x_i)$ converges to zero.

INSTRUCTIONS FOR RUNNING NEWTON
Follow the LOGIN procedure and type NEWTON for the program name. You will be required to give the computer the function and its derivative. If you need help in defining the function for the computer, you can run the program called FNCTN. You will be asked for the initial estimate of the root x_1. A table containing x_1, x_2, x_3, x_4 and x_5 along with the values of the function will be printed. You will have the option of continuing the procedure or starting with another estimate. Instructions are given in the program, but if problems occur refer to the LOGIN procedure.

LESSON 1 - FUNCTIONS

This lesson contains tutorial computer programs which explain the symbols used to evaluate mathematical expressions and functions. A description of functions provided by BASIC is given and several examples are illustrated. Even though the student is not expected to know computer programming, it will be necessary to input mathematical functions. Many of the other programs require that the student write a line of the program which defines the function. One program in this lesson allows the student to define any function he chooses. The computer will evaluate the function for values which he specifies.

LESSON 2 - LIMITS

This lesson should strengthen the student's intuition concerning limits. By having the computer evaluate a function for a large number of values near the point where the limit is taken, the user should be able to see whether the function is approaching a limit and estimate the value of the limit. One program contains seven predefined functions and the student is to examine the limit as x approaches zero. On the basis of functional values he must decide whether each function has a limit, goes to infinity or has no limit. If the function has a limit at zero, he must approximate the limit and choose x close enough to zero to come within a specified epsilon of the limit. If the limit is infinite, the user must choose a value of x so that the function will be greater than a specified number.

Most students have difficulty in seeing the relationship between epsilon and delta in the standard definition of limits. So that a student may get a feeling for the definition, he examines a table containing values of x and values of the function. Using the information he must determine how close x must be to A so that f(x) will be close to L where

$$\lim_{x \to A} f(x) = L.$$

The program allows the student to input values of x and each time he inputs a value a table is printed with the values arranged in numerical order.

LESSON 3 - DERIVATIVES

This lesson demonstrates how a limit is used to evaluate the derivative of a function. The student defines a function and specifies a value, a, where the derivative is to be evaluated. The computer will then find the slope of the secant line through the points (a, f(a)) and (a + h, f(a + h)) for specified values of h. The student examines the slope of the line as h approaches zero and determines an approximate value of the derivative. By observing that for very small values of h the results become inconsistent, the student is able to see the limitations of the computer due to round-off error.

LESSON 4 - EXTREME VALUES

This lesson is intended to introduce the student to a simple numerical method for determining the extreme values of a function on a given interval. It is clear that if the function is evaluated at enough points the extreme values can be determined to any accuracy desired. A program allows the student to input 20 values of x and will give the values of the function at these points. The student may input any number of points at a time before the table is printed and should try to find an efficient algorithm for finding the extreme values. The student inputs values of x, examines the table and from the values given in the table determines the best choice for selecting other values of x. It can be shown that for a continuous function with one relative extreme value in the interval (a,b), the point where the extremum occurs can be determined to within

$$(b-a)(1/3)(1/2)^{(n-4)/2} \text{ with } n \text{ points.}$$

With 20 points a student could determine accuracy to about 1/1000 of the length of the interval.

LESSON 5 - NEWTON'S METHOD

We have already mentioned the one program that accompanies this lesson. This is an excellent example of an application of differentiation used in an iterative procedure. Examples are given to show that the sequence may not converge and to illustrate other problems that may occur. The program lists each value of the sequence and the values of the function at these points. The student is to determine whether the sequence converges to a solution.

LESSON 6 - CONICS

This lesson examines the general quadratic equation

$$ax^2 + bxy + cy^2 + dx + ey + f = 0.$$

The computer program will give the angle of rotation and reduce the equation to a form similar to one of the following types for a conic or the appropriate degenerate case.

$$\frac{(x-H)^2}{A^2} \pm \frac{(y-K)^2}{B^2} = 1$$

$$(x-H)^2 = C(y-K)$$

$$\frac{(y-K)^2}{B^2} - \frac{(x-H)^2}{A^2} = 1$$

$$(y-K)^2 = C(x-H)$$

In order to get the final result the student must answer questions concerning the nature of the graph, and with each correct answer the equation is written in a form more nearly like the standard form.

Examples of questions that the student must answer are:

1. Is the graph of the equation a parabola, ellipse or hyperbola?
2. Is this the degenerate case?
3. Is the graph a point, a circle, a real ellipse or no locus?
4. Is the axis of the parabola parallel to the x-axis or the y-axis.

When the student has answered all the questions correctly he is given a form of the equation that can be easily graphed and given the number of questions that were answered incorrectly.

COMPUTER PROGRAMS

We are currently using a DEC-10 computer and programs are written in PDP-10 BASIC. Most students use the DECSCOPE VT50 terminals which have a BAUD rate of 1200. Hard copy terminals are available but for most of the programs in the series, only a small amount of information need be recorded. It is much easier to read information on the video-terminals and the output rate is greater, so the programs are designed for use with this type of terminal.

SUMMARY

We have offered a one-hour computer laboratory course for the last four years. The laboratory can be taken with any mathematics course. Because of limited enrollment in some of the courses it is difficult to schedule a laboratory class for every course, but the lessons are designed so that with the help of the computer the student needs only minimal help from the instructor. This allows us the flexibility of offering the laboratory class where only 2 or 3 students from a particular course may enroll.

When we first began the program we were batch-processing but there are many advantages using interactive terminals with a time-sharing system. Since we obtained our new computer there has been an increased interest in using the computer. Not all students are computer bugs, but it is important that interested students be given opportunities to use the computer. Many students are enthusiastic about the possibilities the computer offers and seem to have an increased interest in mathematics. Many students become deeply involved and give suggestions or write computer programs which we incorporate into our lessons.

Programs of the type we use are not difficult to write. There are many resources available, and they are an excellent source for ideas. However, we find that it is just as easy to write our own programs. They are designed to fill our particular need, we understand the program, and it will run on our computer.

TOTAL COMPUTERIZATION OF A COURSE

Frank D. Vickers
Department of Computer and Information Sciences
University of Florida, Gainesville

A course at the University of Florida has become so involved with the computer that it can be considered the perfect example of total computerization. Involvement has expanded into every facet of the course to the extent that each activity is either controlled or augmented by the machine. This is even true of the lecture which is strongly supported by a time-sharing video terminal connected to a large screen projector. Other areas of involvement include CAI, CMI and CATC as well as administrative applications. The course, CIS 302, is entitled An Introduction to Computer Programming, is worth two quarter hours and has an enrollment of 800 each quarter. The primary subject of the course is the Fortran language and its applications.

COMPUTER ASSISTED LECTURING

The computer becomes an invaluable and exciting tool in the classroom when it is coupled with the large screen projector and the time-sharing terminal. Lectures really come to life when every point being made can be illustrated with a live demonstration. If a student has a question involving a variation of the point being made an immediate demonstration of the effect can be shown. With the time-sharing terminal operating at a speed of at least 600 baud almost any behavior of the computer can be demonstrated without any significant delay to the progress of the lecture.

Disemination of information such as ID and account numbers and laboratory appointment times for each individual in the class becomes easy when displayed quickly and accurately with the computer. Distribution of such information for 250 people can be easily accomplished in about ten minutes.

The power of the computer really comes home however when the discussion of algorithms or similar topics arise. For example, if the concept of the computed GO TO in Fortran is being introduced, the following program and demonstration can be used to advantage. User input commands and data values are identified by an underscore character in column one.

```
_erase
WORK FILE ERASED
_fetch grader
GRADER   FETCHED
_list
0000       1 READ, GRADE
0001         IG = GRADE/10
0002         GO TO (2,2,2,2,2,3,4,5,6,6), IG
0003         PRINT,'ERROR IN GRADE INPUT'
0004         GO TO 1
0005       6 PRINT, 'GRADE IS A'
0006         GO TO 1
0007       5 PRINT, 'GRADE IS B'
0008         GO TO 1
0009       4 PRINT, 'GRADE IS C'
0010         GO TO 1
0011       3 PRINT, 'GRADE IS D'
0012         GO TO 1
0013       2 PRINT, 'GRADE IS E'
0014         GO TO 1
END OF WORK FILE
```

```
∠fortran
BEGIN COMPILATION       10
000      1 READ, GRADE
78
GRADE IS C
000      1 READ, GRADE
59
GRADE IS E
000      1 READ, GRADE
66
GRADE IS D
000      1 READ, GRADE
84
GRADE IS B
000      1 READ, GRADE
99
GRADE IS A
000      1 READ, GRADE
$stop
    .01 SECONDS CPU TIME 00000 1
PROCEED
```

The presentation and discussion of complex subjects such as formatted input and output in Fortran becomes routine when specific demonstration programs are designed and used. For example an illustration of the behavior of the integer field in Fortran is a case in point. This demonstration clearly illustrates what happens when the integer data values are not right justified in their respective fields.

```
∠erase
WORK FILE ERASED
∠fetch inputi
INPUTI   FETCHED
∠fortran
BEGIN COMPILATION     11

ENTER INTEGER DATA FOR I, J, AND K
FORMAT IS -   I2, I3, I4
COLUMN
12345678901234567890

5 25 97

INTERNAL VALUES ARE:

   50  250 9700

    PAUSE
```

The following demonstration illustrates how real F fields behave. Here the input data values do not contain decimal points so it becomes obvious that the computer points off the number of fractional digits indicated by the field specifications.

```
∠erase
WORK FILE ERASED
∠fetch inputf
INPUTF    FETCHED
∠fortran
BEGIN COMPILATION    12
```

339

```
ENTER REAL DATA FOR X, Y, AND Z
FORMAT IS -   F4.1, F5.2, F6.2
COLUMN
12345678901234567890

_426 7350  8425

INTERNAL VALUES ARE:

       42.6        73.50        84.25

    PAUSE
```

The following illustration shows the behavior of I, F and E fields used in output. Notice how F fields are filled with stars when the value being printed is too large for the field while E fields seldom ever cause difficulty with very large or small values.

```
∠erase
WORK FILE ERASED
∠fetch output
OUTPUT    FETCHED
∠fortran
BEGIN COMPILATION    14

ENTER X, DX AND ICOUNT
FORMAT:   2F5.1, I5
COLUMN
12345678901234567890

25.0 25.0    15

OUTPUT FIELDS:  I4, F6.2, F8.2, E14.4

  I    X     X SQ       X CUBE

   0 25.00   625.00     0.1563E 05
   1 50.00  2500.00     0.1250E 06
   2 75.00  5625.00     0.4219E 06
   3100.0010000.00     0.1000E 07
   4125.0015625.00     0.1953E 07
   5150.0022500.00     0.3375E 07
   6175.0030625.00     0.5359E 07
   7200.0040000.00     0.8000E 07
   8225.0050625.00     0.1139E 08
   9250.0062500.00     0.1563E 08
  10275.0075625.00     0.2080E 08
  11300.0090000.00     0.2700E 08
  12325.00********     0.3433E 08
  13350.00********     0.4287E 08
  14375.00********     0.5273E 08
```

The equipment cost for integrating the computer into the classroom is minimal. A video terminal with the appropriate features will run about $1,500.00 and the video projector is abovut $5,000.00.

COMPUTER ASSISTED TEST CONSTRUCTION

The computer is highly involved in the testing aspects of CIS 302. Four one-hour tests and a final examination are all totally generated by machine. A unique system has been developed which allows the production of objective type test questions which force the student to at least be able to understand existing Fortran program segments. Thus the student is tested at the analysis level of cognitive behavior. Research is still in progress with the hope of reaching the point where the cognitive level of synthesis is measured thereby determining the student's ability to write Fortran programs. The present test generation system is based on a new algorithmic language called TOSL which allows the modeling of each question to be generated. The language allows the definition of a problem in English containing random variability. Operators are provided to build bridges between the English statement of a problem and the algorithmic definitions of the correct and four incorrect answer choices.

The TOSL system can then be used to generate several versions of an examination for application to alternate students within one class section as well as between different class sections. Sample examinations can be published in the course syllabus and are discussed in class before the application of the actual test.

Of course optical-reader answer sheets are used to obtain the responses of the students to the tests. The optical reader then converts the data to magnetic tape for input to the master grading program for integration into the student record data base. Through the use of an intraclass ID (ICID) number as well as social security number the quiz scoring routine makes every possible effort to identify every paper in the presence of errors. Out of 700 papers perhaps only one or two will be beyond recognition. The quiz scoring routine also produces a cross reference listing between student ID and quiz paper sequence. Therefore a paper can be located relatively easily for manual inspection without physically sorting all of the papers by hand. A document sorter is not available.

ADMINISTRATIVE FUNCTIONS OF THE COMPUTER

The application of the computer to the problems of keeping records for a large class becomes very useful particularly if real time access is also provided. The following example of a conversation between the system and a user illustrates several useful operations. First the inspection of record (card) 2006 is requested. The name field is then updated with a new name. Then the score for quiz one and homework one is updated. This is followed by a second inspection of the record. Then, finally, the record is flaged as being dropped from the roll. Other access routes include the possibility of changing the social security number, changing an appointment time, restoring to the roll a dropped student, and finally the ability to add new students to the end of each class section.

```
/swap update
RSCS - JOB OFO BEG EXEC REGION(1) 220K TIME=15.59.08
ENTER # COMMAND
SYSIN
#card
ENTER 4 DIGIT ID #.
2006
266253133 THIS SLOT NOT USED   20061  0  0  0  0  0  0  0  0  0  0  0000000000 E
#name
ENTER 4 DIGIT ID # AND 21 COLUMNS OF NAME.
2006doe john robert
#quiz
ENTER 4 DIGIT ID #, 1 DIGIT QUIZ #, AND 3 DIGIT SCORE.
20061 95
#hmwk
ENTER 4 DIGIT ID # AND 1 DIGIT HMWK #.
20061
#card
ENTER 4 DIGIT ID #.
2006
266253133 DOE JOHN ROBERT      20061 95  0  0  0  0  0  0  0  0  0  0100000000 E
#drop
ENTER 4 DIGIT ID #.
2006
#card
ENTER 4 DIGIT ID #.
2006
266253133*DOE JOHN ROBERT      20061 95  0  0  0  0  0  0  0  0  0  0100000000 E
#end
RSCS - JOB ABND CODE=U0101
RSCS - JOB OFO END
PROCEED
```

341

The entire system is backed up by a sophisticated batch-mode program which performs every conceivable function of record maintenance including the allocation and reallocation of laboratory appointment times for student convenience. Several additional batch-mode programs are used to initialize class rolls and resolve differences between registrar lists and actual student attendance. Another program is being developed to aid in the transcription of final grades to the registrar grade report at the end of the quarter.

COMPUTER GRADING OF HOMEWORK

Students currently are expected to complete eight homework assignments using the computer. More than half of the eight problems are solved using time-sharing terminals. CAI techniques are used to introduce the student to the system. But the unusual aspect of this part of the course is the automatic grading and recording of each problem as the student completes it. Each student also has real-time access to his quiz and homework scores by way of the time-sharing system.

To move the students into the computer environment as fast as possible the first problem solution is given to them. Instructions are presented in a syllabus to provide step-by-step guidance in the process of running the problem. The first problem involves a simple payroll computation, an application that everyone should understand. The CALL CHECK statement provides the link to the automatic scoring and recording system. The following presentation shows the program and conversational interaction during the running of this problem. Again student input is identified by an underscore character in column one.

```
_list
0000      3 READ, HOURS, RATE, TXRATE
0001        WAGE = HOURS * RATE
0002        TAX = TXRATE * WAGE
0003        PAY = WAGE - TAX
0004        PRINT, HOURS, WAGE, TAX, PAY)
0005        CALL CHECK11(HOURS, WAGE, TAX, PAY)
0006        GO TO 3
END OF WORK FILE
_fortran
BEGIN COMPILATION     7
000      3 READ, HOURS, RATE, TXRATE
_40.0, 1.75, .15
HOURS= 40.0000      WAGE= 70.0000      TAX= 10.5000      PAY= 59.5000
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
000      3 READ, HOURS, RATE, TXRATE
_15.0, 1.50, .12
HOURS= 15.0000      WAGE= 22.5000      TAX= 2.70000      PAY= 19.8000
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
000      3 READ, HOURS, RATE, TXRATE
_8.5, 2.13, .17
HOURS= 8.50000      WAGE= 18.1050      TAX= 3.07785      PAY= 15.0271
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
000      3 READ, HOURS, RATE, TXRATE
_48.0, 6.55, .27
HOURS= 48.0000      WAGE= 314.400      TAX= 84.8880      PAY= 229.512
CHECK1 HAS CONTROL
YOUR PROGRAM IS CORRECT.
ENTER YOUR ICID AND SOCIAL SECURITY NUMBER IN FORMAT SHOWN.
ICID,SSS-SS-SSSS
_1014,264-31-8742
SUCCESSFUL COMPLETION OF PROBLEM 1 HAS BEEN RECORDED.
GRADE REPORT FOR BARNES, JENNIFER LEE

   QUIZ  1   2   3   4   5   6   HMWK  123456789    APMT
        88   0   0   0   0   0         100000000      0
CONTINUING...
000      3 READ, HOURS, RATE, TXRATE
_$stop
PROCEED
```

There currently is a flaw in the system since the following program sequence will also be accepted as correct. This problem will be elliminated within a few months when a new system that is currently being implemented becomes available. This new system will run the students program a second time using data which is unknown to the student. Therefore there will be no way he can subvert the system.

```
∠list
0000        CALL CHECK11(40.0, 70.0, 10.50, 59.50)
0001        CALL CHECK11(15.0, 22.50, 2.70, 19.80)
0002        CALL CHECK11( 8.5, 18.10, 3.08, 15.03)
0003        CALL CHECK11(48.0, 314.4,84.88,229.51)
0004        STOP
END OF WORK FILE
∠fortran
BEGIN COMPILATION      8
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
CHECK1 HAS CONTROL
CURRENT ITERATION IS CORRECT.  CONTINUING...
CHECK1 HAS CONTROL
YOUR PROGRAM IS CORRECT.
ENTER YOUR ICID AND SOCIAL SECURITY NUMBER !N FORMAT SHOWN.
ICID,SSS-SS-SSSS
1014,264-31-8742
SUCCESSFUL COMPLETION OF PROBLEM 1 HAS BEEN RECORDED.
GRADE REPORT FOR BARNES, JENNIFER LEE

   QUIZ  1   2   3   4   5   6   HMWK  123456789     APMT
         88   0   0   0   0   0        100000000       0
CONTINUING...
     .01 SECONDS CPU TIME  00000 1
PROCEED
```

CONCLUSIONS

It is perhaps important to note that total computerization does not elliminate the need for enthusiastic and intelligent leadership on the part of lecturers, laboratory assistants and office consultants. The computer integration simply minimizes the tedious time consuming dog-work and helps to maximize the information transfer between teacher and student and provides an exciting and meaningful educational experience. This experience is measurably enhanced by the speed, accuracy, and flexibility of the computer. The amazing thing is the integration can be carried out with reasonable cost. It does require the attention of knowledgeable and enthusiastic people to bring all of the pieces together and make them work in a coordinated way. The one most valuable aspect from an educational viewpoint would have to be the integration of time-sharing and the projection system into the classroom.

COMPUTER-MANAGED INSTRUCTION

CMI FOR STAFF DEVELOPMENT IN A COMMUNITY COLLEGE SYSTEM

Dr. Edward Boas, Jr., Dir. DELTA
Dr. Nevin Frantz, Jr., Professor
Dr. John Matthews, Chairman,
Dept. of Occupational Education
University of Delaware

In response to the staff-development needs of Delaware's four-campus community college system, computer-managed instruction (CMI) proved to be a functional instructional delivery system. Individualized instruction modules were delivered statewide to twenty-six instructors via DECwriter computer terminals. The three-part CMI program titled CREATE/EXPLORE/RECORD managed the instructional delivery by a DEC PDP11/70 computer using Resource Sharing/Time Sharing (RSTS) 6B operating system. It accepted inputs, distributed modules, generated alternative exams, kept records on all transactions, and performed statistical manipulations. The system was developed by the Department of Occupational Education in conjunction with its DELTA Educational Computing System.

INTRODUCTION

During 1974-75 the authors became interested in the usefulness of the computer for its text editing and storage capability as well as its inherent management ability. Because CAI was so terminal intensive, it was determined that an instructional delivery system was needed that could assist an instructor, manage the instruction, keep records, keep track of needed resources, and manipulate item analysis and statistics. A new computer managed instruction system seemed the best avenue to follow. Using a small research grant, and the services of two ingenious programmers, Mr. Gary Luckenbaugh and Mr. Walter Mahla, from DELTA Computing System, Dr. Boas accepted the task to develop and pilot what became a rather sophisticated CMI program. It was written in three parts using the BASIC-PLUS language which could operate under the Resource Sharing/Time Sharing (RSTS/E) operating system for up to 63 simultaneous users. The DELTA Computing System was in statewide operation.

The EXPLORE CMI system became the basic departmental delivery program for its computer-based instructional delivery. Piloting and revision of the programs evolved through 1974-77 with great promise. Teachers were beginning to use individualized modules in classes and those who were in DELTA member schools found the system to be fun and useful.

COMMUNITY COLLEGE STAFF DEVELOPMENT

Instructors at technical and community colleges are usually employed as a result of their expertise and experience in business, industry, or public service. Many of these individuals have little or no preparation in planning and delivering instruction for post-secondary students. In addition, many instructors are employed on a part-time basis. This intensifies the problem associated with developing a coordinated and consistent approach to most departmental instructional programs. The usual one day staff-development workshops and preservice orientation sessions have not been satisfactory approaches in alleviating many of the problems instructors face in developing and teaching their courses.

During the past two years a different approach to staff development has been cooperatively developed by the University of Delaware and Delaware Technical and Community College. The approach utilizes a computer-based system to deliver and manage instruction.

Delaware Technical and Community College, which is located on four campuses within the State of Delaware, began the staff development program two years ago. Approximately ten instructors were selected from each campus to participate in a series of six mini-courses dealing with instructional systems design. These one-credit graduate courses were conducted throughout the year on each campus by a faculty member from the College of Education of the University of Delaware. Each mini-course was completed by the participants using a self-instructional module. They also attended class sessions which were held to discuss and clarify information and constructively critique the instructional materials under development. At the end of each mini-course, the participants were evaluated on their finished products and the cognitive knowledge of the mini-course subject.

During the final mini-course in the spring quarter, the participating instructors were required to field test the instructional unit or module they had prepared using the knowledge gained from the preceding courses. Each instructor then prepared: (1) a pre-check, (2) a terminal-performance objective, (3) a learning sequence of enabling objectives, (4) a mediated instructional resource, (5) learning activities with feedback for each enabling objective, and (6) a postcheck to evaluate student achievement. Many of the instructors formatted the unit in a self-instructional module while others used a lecture-discussion approach. The results of the field test were documented, by each participant, in a written report. This was shared orally with class peers, the deans of instruction, and central administrative staff during the final class session.

During the first year of operation, several logistical and instructional problems occurred in delivering and managing the instruction provided to the participating instructors. The logistics of writing, typing, reproducing, collating, and distributing quantities of self-instructional modules to participating instructors located on four different campuses created problems throughout the year. Quite often instructors who had completed a module had to wait until another instructional module could be printed and distributed before they could begin to work on the next mini-course. Any changes in the module, due to typographical errors or modifications to instructional activities, had to be retyped and reprinted before they could be used again with another group. This proved to be an expensive and time consuming process.

During the last class session of each mini-course, a twenty-five item postcheck was administered to each instructor to determine knowledge of the principles and concepts involved in the course. Because of demanding class schedules and other commitments, instructors in the mini-courses were not always prepared to take the postcheck during the last class session. In addition, the approach utilized valuable class time which could have been more productively spent in additional discussion if another alternative had been available.

IMPLEMENTATION OF COMPUTER MANAGED INSTRUCTION

As a result of these logistical and instructional problems, an analysis was made of existing alternatives that could be utilized to deliver and manage instruction for individuals located on each of the four campuses of Delaware Technical and Community College. One of the readily available alternatives was to deliver instruction via an existing statewide computer network which linked three of the four campuses. The network, provided to DTCC under a contract with the University of Delaware Computing Center, was initially developed for student and administrative record keeping purposes. After consulting with various individuals at both institutions, it was decided to refine our computer managed instruction program which would use a PDP 11/70 computer to (1) store and deliver a series of self instructional modules, (2) provide postchecks, and (3) monitor student performance on each of the modules. During the first six months of 1977, the necessary programming was developed, a data base was prepared, and a computer terminal installed at the fourth campus.

FIELD TESTING AND REVISIONS

The computer-based instructional delivery and management system was field tested in the Fall of 1977 using a group of twenty-six instructors representing various campuses of Delaware Technical and Community College. Each instructor was given an orientation to the computer system and directions for accessing the instructional modules. Instructors using the system selected a terminal-performance objective for an instructional module and had an option of taking a precheck which measured their prior knowledge of the information found in the module. After completing the precheck off-line, their responses were entered via the terminal into the computer, which analyzed the information and printed out the appropriate instructional sequence of enabling objectives. Instructors not electing a precheck were given all of the enabling objectives required for the instructional module.

Each enabling objective on the computer print-out had a list of available instructional resources, several student activities, and a feedback section for verification of correct responses as well as directions for remedial instruction. After successfully completing the activities for each enabling objective, the instructors obtained the computer generated postcheck. It consisted of twenty-five multiple-choice items which were written to measure comprehension, application, and synthesis of the concepts and principles presented in the module. The postcheck was printed out, completed off line and responses were entered using the terminal keyboard. If criterion performance (20 items correctly answered) was achieved, feedback on any items answered incorrectly was given. Results were recorded by the computer, and the instructor could begin another instructional module. However if criterion performance was not achieved, the number of items answered incorrectly was provided to the instructor with directions for locating the correct information in the listed resources. Upon completion of the remedial route, an instructor would obtain and complete the second postcheck. Feedback provided to the instructors after completing the second postcheck included the number of correct responses, incorrect items with their correct answers, and a short narrative indicating why the answer was correct.

RECORD KEEPING AND MODULE REVISION

The ongoing status of each instructor was also recorded and stored by the computer. Information about instructional modules completed, postchecks attempted, and individual scores were available on computer printouts. In addition, information was also collected on items answered incorrectly and responses selected by those who missed the item. The collected data was used to develop a profile of each module, which was analyzed in order to modify and improve the instructional system.

The results of the field test indicated that using computer-based technology was a viable approach to delivering and managing instruction for a group of individuals located on a multi-campus college. Several of the logistical and managerial problems were quite readily resolved using the system.

The problem of modifying or changing the modules after idscovering typographical errors or poorly designed instructional activities was accomplished with relative ease in a short period of time using the computer-based text editing system. In fact, problems uncovered by initial users of the modules were corrected before most of the remaining instructors began to work on the modules. Under the paper-packaged system this would have been impossible and could only be done for a new group in the following year. Under the computer-based system, instructors could also complete and begin another instructional module at their convenience as opposed to a time dictated by printing and distribution of schedules.

The use of a computer-managed approach assisted in the process of individualizing instruction by preassessing entry behaviors and prescribing appropriate enabling objectives for each of the instructors. Instructors were also given the opportunity for self pacing since the postchecks were always available under the computer system, as opposed to being administered at a given time for all individuals. The record keeping system provided the coordinators of the program with a method of easily determining and monitoring the individual progress of each instructor. Data collected from the pre and post checks was easily retrived and was useful in making improvements and modifications to the postchecks, instructional resources, and activities within the modules.

FEEDBACK

Initial feedback from the instructors concerning the use of the computer-based system ranged all the way from negative to enthusiastic during the initial stages of the field test. For many of these individuals, however, it was their first encounter with a computer and several programming and technical problems did little to enhance their initial experience. As these problems were eliminated, however, and the instructors became accustomed to interacting with the system, their attitudes improved and became more positive. These positive attitudes were displayed in many ways. Several of the instructors began to attend special "computer orientation" workshops held on their campuses while others enrolled in various university and college computing-applications courses. Throughout the year, several instructors began to use the system in delivering and managing instruction for their own students.

THE COMPUTER MANAGED INSTRUCTION PROGRAM

The actual computer program to operate the Computer-Managed Instruction system (CMI) was developed by the Department of Occupational Education, University of Delaware, in conjunction with one of its component programs, the DELTA Educational Computing System. The CMI program actually delivers and manages individualized instructional modules using three interacting programs: CREATE, EXPLORE, and REPORT.

The three CMI operating programs were written in Digital Equipment Corporations's BASIC-PLUS language. They run under the Resource Sharing Time Sharing operating system (RSTS/E 6B) on the DEC PDP 11/50 or 11/70 computers. Modules can be delivered by phone lines to either video or printing terminals. Up to 63 users may participate at one time.

CREATE

The CREATE program is the computer-managed instruction system (CMI) manager's tool. He uses it to take the instructional modules developed by instructors and properly build them for the EXPLORE program to use. Using this program, the system manager can also specify what modules are on-line and off-line at any time. He can also take a specific module off line to update it without affecting any other modules in that area or cluster. The system manager can also limit access to certain modules by specifying what user codes may access that module or area or cluster. Another option of the CREATE program is a bypass command. This allows one to bypass all parts of the module except for the post check. In this way, the EXPLORE program can be used as a method of delivering a final check or exam on most types of instruction.

EXPLORE

The EXPLORE program delivers the instructional modules to the student. The student signs on the system and is given a list of the clusters available. After choosing a cluster, the list of areas available in that cluster is printed out. The student then chooses the area he wishes to work on. Following this choice, the student is given a list of the modules available in the chosen area. The list describes the terminal performance objective for each module. The student may also get a print-out of the complete TPO for any of the listed modules. After selecting the module he wishes to work on, the student is asked if he would like to take a pre-check. If so, the pre-check is printed out on the terminal. The student would then sign-off the system and go back to class to work on the pre-check. After completing the pre-check, the student would come back to the computer and access the EXPLORE program again. This time the student would input his answers to the precheck. Based on these answers the EXPLORE program gives the student the necessary learning packages of the module. If the student chooses not to take a pre-check, he is given all the learning packages of the module. After the module has been printed out on the terminal, the student would sign off the system and go back to class to work on the module. After completing the module, the student would come back to the terminal and access the EXPLORE program again. This time the student would be given a post-check for the module just completed. After the post-check has been printed out on the terminal, the student would sign off the system and go back to class to work on the post-check. After completing the post-check, the student would come back to the terminal and access the EXPLORE program again. This time the student would input his answers to the post-check. The system then tells him whether or not he has passed the module. If the student has not passed the module, he is told either to come back and review the TPO at a later date or is given remedial feedback and the opportunity to take an alternate post-check. After reviewing the remedial feedback, the student then takes the alternate post-check following the same procedure that he did for the first post-check. If he does not pass the first alternate post-check, he is given more detailed remedial feedback and then given the opportunity to take a third and final post-check. After completing all the post-checks, if the student has passed the module, this is recorded. If the student has not passed the module, this is also recorded; and the student is told to come back and review the TPO at a later date. At this point the EXPLORE program is concluded.

REPORT

The REPORT program is the instructor's gradebook. The program keeps track of the students using the EXPLORE program, what module they are working on, at what point they are in the module (i.e., pre-check, instrucitonal packages, post-check, first alternate post-check, or second alternate post-check), the time and date the student started working on the module, and the time he last worked on the module. The REPORT program will also print out a detailed listing of the student's answers to the pre-check and post-checks. This feature is like an item analysis. The REPORT program also enables the teacher to add or delete from the list of those students who are using the EXPLORE program. It also allows the teacher to specify what students are qualified to take the alternate post-checks.

CONCLUSION

This is a very efficient and easy to use CMI system. Its format and operation make it possible to adapt most types of instruction for use on the system. Through TECO, a simple text editing program, teachers can enter their own modules and then request the system manager to add them to the proper cluster. In this way, a large and varied selection of modules is made available in many subject areas.

The advantage of the CMI system for teachers is that it provides the basic instruction to the students and frees the teacher to spend more time working with those students who may need special help or intense individual instruction.

A GENERALIZED SYSTEM FOR CLASSROOM INSTRUCTIONAL MANAGEMENT

Charles J. Clock, Jr. and
Gail K. Witt
West Hartford (Connecticut)
Public Schools
Abby Pinard - National CSS

The purpose of this paper is to describe the activities involved in using a commercially available generalized data base management system for monitoring pupil progress in a K-3 mathematics program. Some school districts fail to consider the development of computer based classroom management systems because they do not have the hardware or technical expertise necessary to support the task. The generalized data base management system called NOMAD, available on a time-sharing basis through National CSS, was used to develop the instructional management component of the math program described in this project. Major advantages of using this commercially available service were found to be in the simplicity and flexibility of report specification and generation, generalizability of application, low cost of operation, and fast turn-around capability.

The Primary Mathematics Management System (PRIMS) was a pilot program in the West Hartford, Connecticut, Public School system during the 1975-76 and 1976-77 school years for children in Kindergarten through grade 3. Local educators felt effective procedures for monitoring the progress of primary level children were not currently available through commercial materials. The purpose of PRIMS was to respond to this need by providing a simplified procedure for monitoring the progress of young children in mathematical concepts and objectives. PRIMS was composed of: 1) a scope and sequence of math skills reflected in most commercial elementary math programs; 2) a set of diagnostic tests for assessing skill proficiencies and deficiencies; 3) a simplified procedure for administration and scoring diagnostic tests, and 4) the use of a commercially available generalized data base management system for processing and reporting pupil progress to teachers and parents.

Figure 1 shows the organization of the Primary Mathematics Management System.

The following are more detailed explanations that refer to each phase of the management system cycle:

Description. The total management system is organized around major mathematical concept areas for levels K-3. Each concept has performance objectives that specifically relate to it.

Checkpoints. A series of 11 Comprehensive Checkpoint Tests has been developed with items that measure each objective in the Primary Mathematics Management System. The checkpoints are administered to students at levels K, 1, 2, and 3 up to three times a year, according to student levels in mathematics.

Computer Process and Storage. The tests are hand scored and the item data are transferred to machine scorable answer sheets. A computer program transforms the item data to skill mastery level information according to pre-specified criteria. Item data are retained for future item analysis purposes and the skill mastery data are input to NOMAD for report generation. Skill data are stored as history information.

Student Performance. Information pertaining to student performance on the checkpoint tests is then sent to each school. Such information includes both individual and group data. Individual data consists of a print-out for each child listing what the student knows, what objectives the student needs to review, and what objectives the student needs to learn. The individual report lists the objective code, description of objective and the student's performance level on each objective.

Group data consists of several possible reports. These reports are listed as follows:

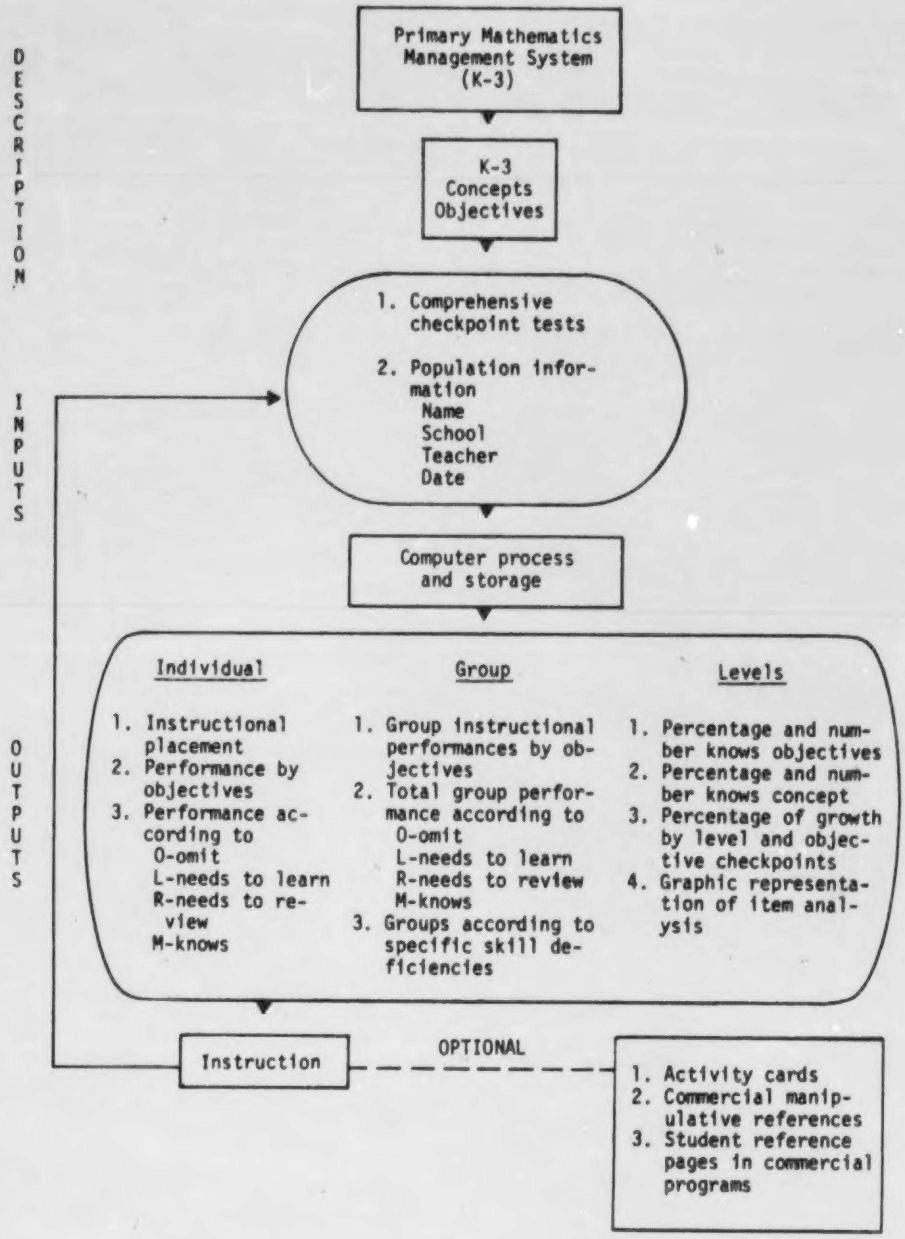

DESCRIPTION
Primary Mathematics Management System (K-3)
K-3 Concepts Objectives
INPUTS
1. Comprehensive checkpoint tests
2. Population information
Name
School
Teacher
Date
Computer process and storage
OUTPUTS
Individual
1. Instructional placement
2. Performance by objectives
3. Performance according to
O-omit
L-needs to learn
R-needs to review
M-knows
Group
1. Group instructional performances by objectives
2. Total group performance according to
O-omit
L-needs to learn
R-needs to review
M-knows
3. Groups according to specific skill deficiencies
Levels
1. Percentage and number knows objectives
2. Percentage and number knows concept
3. Percentage of growth by level and objective checkpoints
4. Graphic representation of item analysis
Instruction
OPTIONAL
1. Activity cards
2. Commercial manipulative references
3. Student reference pages in commercial programs

Figure 1. Organization of the Primary Mathematics Management System

Group Report for each class shows a matrix of all objectives along the top of the sheet, all student names in that particular teacher's mathematics group along the side of the page, and the performance level under each appropriate column.

Objective Deficiency Report for each class, that lists the objective code, description of the objective and all student names in that particular teacher's mathematics group who need to learn or review each objective.

Objective/Level Report for each level that lists objective code, number of students who know that objective on the basis of test scores, and that percentage.

Concept/Level Report for each level, that lists concept, number of students who know that concept on the basis of test scores, and that percentage.

Pre-Post/Level Report for each level, that lists objective code and percentage of growth between two or three testing periods.

Histogram for each level, that diagrams student results for each concept area. Copies of the individual and group reports will be available during the presentation.

Initial problems in the development of the math management system dealt with scoring and reporting procedures. The developers of the program did not want primary level children responding on machine scorable answer sheets, and yet there was a commitment to use the computer for data analysis, storage and reporting purposes. Consequently, it was decided that children would respond directly in consumable test booklets and their responses could be transferred by hand to machine scorable answer sheets. This procedure greatly facilitated the logistics of testing in the classroom by eliminating the need for developing or monitoring mechanical test taking skills of primary level children.

Since this was a new testing procedure which involved criterion-referenced measurement and objective level reporting, it was not clear just what type, format or quantity of data would be most useful. Several initial reports and report formats were proposed for a trial run. Each involved developing computer programs for processing the data. If the teachers or administrators using the reports were to express a need for change, additional programming time and effort would be required. The most efficient way to develop the overall management system was found to be in using an existing generalized data base management system. This would allow for the greatest degree of flexibility with minimal amount of programming involvement. The NOMAD system offered through National CSS was felt to be the most desirable due to its local accessibility, local card input capability, ease of use by a non-programmer user, and relatively low cost of processing and storage. The specifications and programs for an original set of four reports were developed using NOMAD, and a data base was established within two weeks. The West Hartford Public Schools' Computer Center developed a computer program to accept item response data from the machine scorable answer sheets and punch cards containing object level data. Since most objectives are measured by more than one item, the objective level data are determined by the percentage of items answered correctly. The objective level data are established in the following manner:

M = Mastery - all items for a given objective are answered correctly.

R = Review is needed - at least 50% of the items for a given objective are answered correctly.

L = Needs to learn - less than 50% of the items for a given objective are answered correctly.

O = Omit - all items for a given objective are omitted.

The cards produced from the scanned answer sheets contain the "M", "R", "L", or "O" information for each objective. These data serve as input to NOMAD. Tests from a class of approximately 25 pupils are hand scored, cards produced, data processed and reports generated through NOMAD, and final reports in the hands of the classroom teacher within a day. Data from individual pupils or small groups are fed into the NOMAD system via computer terminal and instantly processed for immediate feedback. All data are maintained in history files for future update or longitudinal analysis. The cost of generating the individual and group reports for a class of 25 pupils is approximately $.85 per pupil. However, this figure will vary depending upon the number of pupils processed and whether the reports are generated in an on-line time-share or overnight batch mode.

There are numerous advantages to using commercial time-sharing services. The immediate advantage is that the basic programming already exists and is available to anyone having access to a telephone and a computer terminal.

Consequently, the need for computer hardware or in-house programming talent is unnecessary. Report formats can be tailor made to a school system or a particular teacher with a minimum of effort and response time. The same degree of flexibility can be provided for parent conferencing by adapting the reports to meet specific parent/teacher/pupil needs. This high degree of flexibility, which is part of most generalized data base management systems, also greatly facilitates the formative process of instructional program development since immediate changes can be made in the content or format of the reporting system based on direct teacher feedback.

A major problem in using criterion-referenced measurement is the frequent inability to provide immediate feedback of pupil response data. Commercial test scoring agencies are slow to respond, and this is complicated by the necessity of using mail services to transport both the data and the reports. The telephone provides a much faster means of delivery and opens the door to other users on a regional or national basis.

Another advantage of using commercially available time-sharing services is the accessibility of other on-line programs for analyzing data. Data which are maintained in the system can be further analyzed by using existing statistical packages or item analysis routines. The data can also be displayed and summarized in a variety of ways by using graphic or display programs which are generally available through the same computer system.

The NOMAD system of National CSS was found to be an extremely useful component of the Primary Mathematics Management System. NOMAD provided the flexibility needed to develop reporting aspects of the management system and the immediate feedback essential for effectively monitoring pupil progress. This project has been a practical demonstration of how the public and private sectors can work together to bring an economical solution to an educational problem. This is a valuable resource which needs to be further explored by the educational community.

DESIGN AND IMPLEMENTATION OF AN ELEMENTARY COMPUTER INTERACTIVE READING PACKAGE AND SPELLING PACKAGE

Sharon Conlan-Independent School
District 621, Mounds View
Public Schools - St. Paul, MN.

This paper describes two computerized self-paced instructional packages; spelling and reading. These two packages are based on a planning and evaluating model and the approach employs principles of curriculum planning, instructional design and modern day technology. This paper also deals with why such a system was developed, the development process, and how it is used.

Background

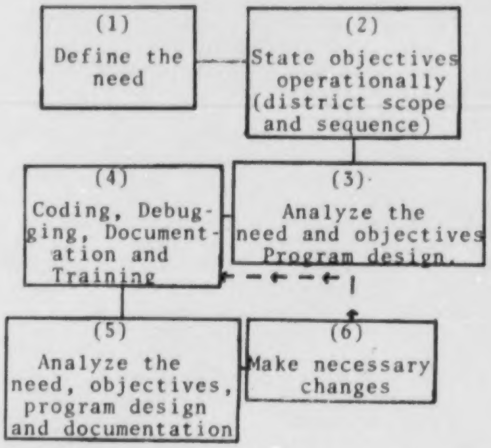
For several years now our school board has had a district goal to develop a comprehensive reading and language arts program for grades K-12. The district has a full time Reading Resource teacher and a fill time Language Arts Resource teacher. These two people have actively been involved in developing scope and sequence for their respective fields, recommending a series of texts for schools to choose from as a learning behavior media, and suggesting teaching strategies for instruction of the scopes and sequences.

The teaching strategy is where I get involved. Having learned of their efforts, I looked into the computer instructional packages available on our MECC system (Minnesota Educational Computer Consortium.) There was no supportive material available. I looked into developed packages for purchase, and either they hadn't been developed following a scope and sequence or they would have required a costly purchase and extensive programming modification. Neither of these alternatives were desirable.

I presented the idea to our school board that I should develop a computerized self-paced instructional package in reading and language arts and that I would pilot these packages in two elementary schools. This meant adding terminals. They unanimously agreed to support such a pilot since it fit in with the district's goals.

Development

Many of us follow some sort of model when we develop something. The one I worked with was as follows:

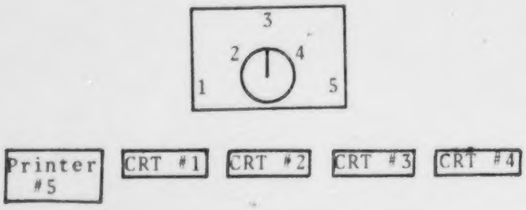

(1) Define the need
(2) State objectives operationally (district scope and sequence)
(3) Analyze the need and objectives Program design.
(4) Coding, Debugging, Documentation and Training
(5) Analyze the need, objectives, program design and documentation
(6) Make necessary changes

Step 1, "Define the need", had been done by the School Board when they stated "... based on test scores our district needs a comprehensive reading and language arts program for K-12." Step 2, "State Objectives Operationally", was also done by the district's Language Arts and Reading Resource teachers when they developed the scope and sequence for those areas. Step 3, "Program Design", involved several steps and several people of different expertise. I studied the design of other computer interactive instructional programs and met with elementary district terminal users for design ideas. At the same time this was going on, letters had been sent to publishers requesting permission to computerize their material. Once the permission had been granted the design was drawn up. First the system design, then the individual program designs.

What equipment to buy was an essential part of the program design. The teachers in our district had used hardcopy terminals more than video display terminals. They were used to having students bring back to them the printout of the work they had been doing on the terminal. I felt this was time consuming on their part, a waste of paper, and could be eliminated if the reports were designed with enough information to tell the teachers of the student's progress.

I convinced the teachers from my two pilot schools that if they had a printer with which video display terminals could send their printout to (one at a time and only when needed) that they would not lose the ability to monitor the student's progress because their teacher obtained student reports would contain the information they needed.

A very important consideration that entered into the decision as to which features should be included in the terminals was the students' ability. Elementary students, as a rule, have never typed but more than likely have been exposed to a calculator. Since the READING package would consist of multiple choice answer selections, I decided the answers they would key in would be a number answer because it would be easier for the student. So with that in mind, numeric keypads were ordered to be included as part of the keyboard. I also ordered cursor control and upper and lower case characters for future expansion. The last extra feature ordered was composite video on one of the video display units. This would enable the teacher to hook a terminal up to a T.V. monitor for classroom or group demonstrations. It was decided to put four video display and one hardcopy terminal in each of the two pilot schools. The hardcopy would have the ability to run as a terminal or, with the turn of a dial, could be a printer for any one of the four CRT's. Below is a diagram of that set-up.



Step 4, 5 and 6 of the model is a reoccurring cycle if the project is to succeed and stay up-to-date year after year.

Documentation and training are essential. The first training session was with all the teachers whose students were going to be part of the pilot. The documentation was gone over page by page with teachers. Then the teachers sat at the terminals, registered some students, took some student lessons, got student reports and edited student records.

Steps 5 and 6 are where we are currently at as we near the end of our first year in using the program. We fully expect and plan on going through steps 4, 5 and 6 each year.

Spelling Package

The district has two approved spelling texts. "Basic Spelling" is published by Lippencott and written by Theodore Glim and Frank Manchester. "Basic Goals in Spelling" is published by Webster Division, McGraw-Hill and written by William Kottmeyer and Audrey Claus.

Four main programs were written to support the two spellers; (1) INITREG, (2) SPELL, (3) EDIT, (4) REPORT. INITREG is used by teachers to register students into one of the two spelling texts and into a teacher-placed starting level.

SPELL is the student session on the terminal. First a student takes a placement test (starting at the level assigned by the teacher) and, based on his/her score, is placed at a level for his/her ability. At the end is a placement test flow for each spelling text.

After the teacher has taught the lesson, a student uses SPELL to take a lesson test. Based on the mastery percentage, entered previously by the teacher, the computer advances him/her one lesson or leaves him/her in the same lesson until mastery is reached.

EDIT is used by teachers to change student level, lesson and name, and to delete a student.

REPORT consists of four different reports. One lists the words missed, one lists placement levels of the students, one lists the current lesson, unit and book, and the last lists the raw scores for the current lesson plus nine previous lessons. All reports can be printed for a group, a range of students, or for a single student.

Reading Package

This package uses what is called a strand structure. A strand is a sequence of related items. There is one strand for each of the four areas covered by the program. The strand approach allows each student to move at his own pace. The material a student receives from one strand is independent of his/her position in other strands and of the progress of other students.

A list of the strands follows, showing the subject area each covers.

Strand A -	Decoding Skills-structural analysis; suffixes, prefixes, root words, compounds and contractions
Strand B -	Comprehension Skills-topic, detail, main idea, sequence and bias

Strand C - Vocabulary Skills - synonyms, antonyms, metaphors, and similes
Strand D - Reference Study Skills - alphabetical order, dictionary skills, classifying words

A student can be in the four strands at all different levels based on his/her individual learning abilities.

Four main programs were developed for this package and were based on a district approved textbook series with publisher's permission. That book is Houghton Mifflin's Reading Series by William, Durr, Jean LePere, and Ruth Brown and is used for Strands A, B and C. Strand D is not developed yet.

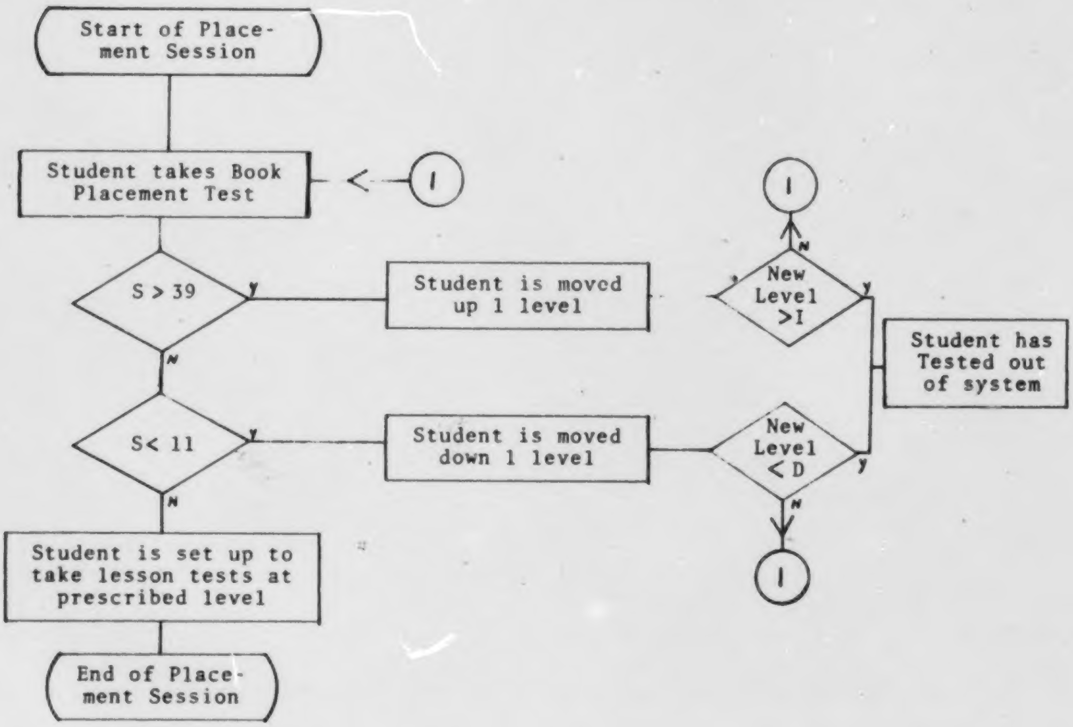
The student takes a placement test for each strand and based on some internally set evaluation, students are raised a level, remain in the placed level or move down a level. Once the level has been determined in each strand and the student sits down to take a lesson, she/he has to key in A, B, C or D depending on the strand she/he is to work on. The computer drills them on that keyed-in strand selection at the level the student had tested into. Each strand is independent of one another and a student may progress at his/her own speed up or down in any strand.

Evaluation

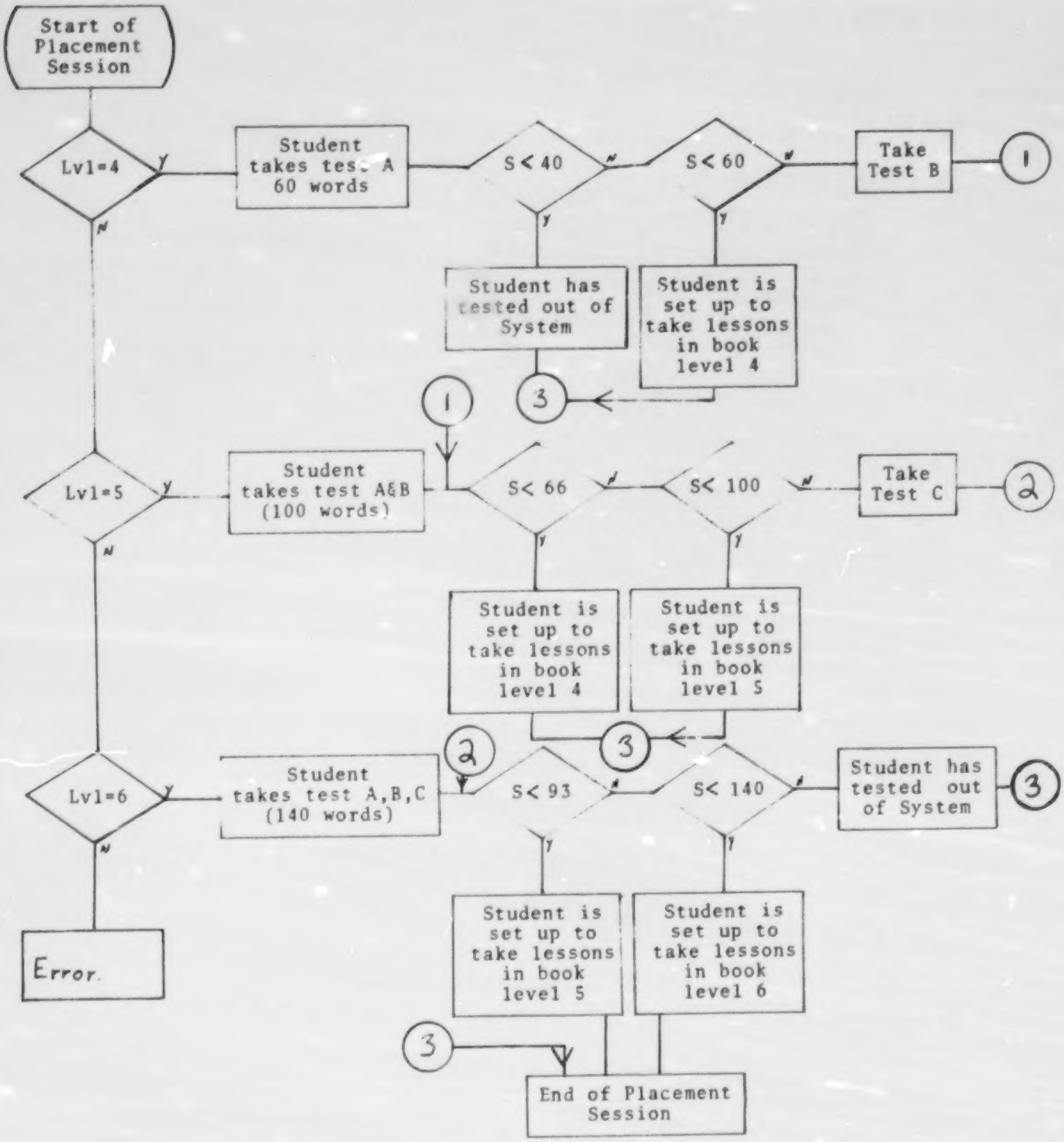
Evaluation all too often is overlooked. We hear people say that students learn better and more quickly on a terminal. Do they really? We'd like to know for sure.

The plans for evaluation involve a control group and an experimental group. Both groups will be in the same text and both groups will keep raw scores on the tests. Our district has its own developed measurement test which the before program exposure scores and after program exposure scores of students will be part of the evaluation. We are currently in the middle of this stage.

BASIC SPELLING


Start of Placement Session
Student takes Book Placement Test
1
S > 39
Y
N
Student is moved up 1 level
S< 11
Y
N
Student is moved down 1 level
Student is set up to take lesson tests at prescribed level
End of Placement Session
1
N
New Level >I
Y
Student has Tested out of system
New Level < D
Y
N
1

BASIC GOALS IN SPELLING


Start of Placement Session
Lvl=4
Y
N
Student takes test A 60 words
S < 40
N
Y
S < 60
N
Y
Take Test B
1
Student has tested out of System
Student is set up to take lessons in book level 4
3
1
Lvl=5
Y
N
Student takes test A&B (100 words)
S< 66
N
Y
S< 100
N
Y
Take Test C
2
Student is set up to take lessons in book level 4
Student is set up to take lessons in book level 5
3
2
Lvl=6
Y
N
Student takes test A,B,C (140 words)
S< 93
N
Y
S< 140
N
Y
Student has tested out of System
3
Error.
Student is set up to take lessons in book level 5
Student is set up to take lessons in book level 6
3
End of Placement Session

COMPUTER SUPPORTED PRECALCULUS:

SOLVING SOME PROBLEMS IN RECORD MANAGEMENT AND TEST GENERATION

Mike Folk
Drake University

A record keeping and test generation system for a modified PSI-type course in precalculus mathematics was developed. Important features of the system include: (a) the relative ease of its implementation, use and alteration because of the use of a high level interpretive programming language (APL) and modular system design, and (b) the use of a teleprinter with limited graphics capabilities for generation of two-dimensional mathematical expressions.

Over the past few years, we have witnessed an unusual amount of innovation in the teaching of precalculus mathematics. Much of the change has been stimulated by the success of a method of instruction known as the Personalized System of Instruction(PSI) or Keller Plan. PSI implies a course structure in which students proceed at their own pace through a series of units, or modules, which they are required to master in succession. There need be no regular lectures since self-instructional materials are usually used, and a large number of student proctors are available to provide help and evaluate progress.

The Mathematics Department at Drake University has, since 1972, been developing a precalculus mathematics program which is similar in many ways to PSI. It is based on the principle that students should have available one-to-one instruction when needed, and that they should be evaluated frequently so that they, as well as their instructors, are aware of how well they are mastering the material in the course. Our courses differ from PSI only in that we do have lectures (one per week), and we require students to complete each unit within one week.

Since PSI-type courses require frequent testing, it is important to have the capacity to generate different, though equivalent, tests on each unit. The problem of generating many different tests has been solved in various ways, including the use of computers to construct tests. In mathematics, the problem is compounded by the fact that many test items should appear in a two-dimensional format (e.g. subscripts, superscripts, fractional expressions, and radicals), rather than the linear format require by most printing devices. Also, unless multiple choice tests are to be used, answer keys should be generated in mathematically acceptable form as well.

PSI courses also typically require much more complex systems of record keeping and grading than traditional courses do. Again, the computer is a valuable tool, and has been used to facilitate record keeping in a variety of ways.

Finally, it is generally desireable to modify easily any system of instruction, either in response to changes in the student population, subject matter emphases or course structure, or simply to facilitate and encourage the evolution of more effective instruction. In the attempt to make computer-based systems operate efficiently (in computer terms), this flexibility is often compromised, if not sacrificed altogether.

The test generation system and record management system described below was designed and implemented with these problems in mind. It reflects a belief that computers have attained sufficient speed and memory capacity to allow programmers to give up some computing efficiency in favor of ease of implementation and alteration, and simplicity of use. A very high level interpretive programming language (APL), together with a modular system design, have been used to accomplish this.

The generation of graphically acceptable test copy is accomplished through the use of a teleprinter with limited graphics capability; it is also programmed in APL.

TEST GENERATION

The test generation component may be divided into two overlapping phases. First, the item bank must be built and maintained. Then, tests are constructed by selecting appropriate subsets of questions from the item bank and typing them under computer control.

Item bank

The item bank consists of a set of files, one for each question type. If there are n valid question types for a given unit, then n files would be assigned to that unit. File names reflect the course, unit, and question type. E.g. file T060305 contains question type 5 associated with Unit 3 of the course Math 6. Reassigning the question type to a different unit or course is accomplished merely by changing its file name.

Files are structured in such a way as to maximize flexibility in the way questions are generated. The first record (record 0) of each file contains a procedure (or identifier of a procedure) which generates the actual question. By allowing the use of different generating functions for different question types, totally different approaches to question generation are possible within the same system.

Other records contain further information to be used in generation of questions. For instance, the most common file structure used has the identifier of procedure FNV in record 0. FNV assumes that each version of the question type has the same "leader," with only minor variations. Record 1 contains a list of numbers used to ensure that questions of this type are chosen in random order, with repetitions of the same version unlikely. Record 2 contains the leader (e.g."Solve for x:"). Records 3 through n+3 contain the text of the n actual versions that have been stored. Each of these records also contains information about variables in the leader, plus solutions, if they have been entered.

```
I

15 4 13 10 2 9 3 14 11 12 6 7 15 8 5 1

Write  the numerator which makes
the fractions equivalent:

8[x@x - 3y]  =  [ @[x^3 - 27y^3]]8

8[y@x + 4y]  =  [ @[x^3 + 64y^3]]8

8[4@x - 2y]  =  [ @[x^3 - 8y^3]]8

8[5@x + 5y]  =  [ @[x^3 + 125y^3]]8

8[x@x + 2y]  =  [ @[x^3 + 8y^3]]8
```

Figure 1. First seven records of a question file.

The greatest problem faced in designing a test generation system was the need to generate mathematical expressions. Most mathematics test generation systems produce either linear FORTRAN-like expressions which students (or a typist) must translate, or two-dimensional expressions which approximate only very crudely the desired mathematical expressions. By using a teleprinter with limited graphics capabilities (Diablo Hyterm II) and a simple mathematical typesetting language which was designed specifically for test generation, it was possible to generate expressions which are at least as good as those done by typists. Only a few symbols must be entered by hand.

The typesetting language looks very much like a mathematical language, but its operators are graphical operators and its operands are rectangular pictures. For instance the operator "@" means "over," so the operation "a@b" produces $\frac{a}{b}$.

"a" and "b" would be operands (rectangles), each with height 1 and width 1. The operator "^" indicates "exponentiation," which is interpreted graphically as "raise the right operand so that its lower fourth is below the top of the left operand and to its right." Thus "x^2" produces x^2.

The result of an operation is a new operand (rectangle), with new height and width. For instance, the result produced by "x^2" has height 1.75 and width 2.

The typesetting language can be altered easily since it is written in APL. Thus far, the syntax includes most symbols used in intermediate and college algebra. As the question bank is increased, the syntax is being extended to meet the graphical needs which arise. Some sample expressions are shown in Figure 2.

input	output
x^2	x^2
*[x^2]	$\sqrt{x^2}$
*[:3:x^[12]]	$\sqrt[3]{x^{12}}$
[1 + [1@x]]@[1 - [1@y]]	$\frac{1+\frac{1}{x}}{1-\frac{1}{y}}$
[x@y] = [@x+y]	$\frac{x}{y} = \frac{}{x+y}$
$av[x+y@x-y]@[1@3]$av	$\left\lvert\frac{\frac{x+y}{x-y}}{\frac{1}{3}}\right\rvert$

Figure 2. Sample expressions generated by the typesetting language.

Test Construction

Given the structure of the item bank and the typesedding language, test generation is straightforward. On a ten question test, the test generator types a header, randomly chooses ten different questionsfor the unit, then successively calls the procedures associated with the questions to format and type them. If an answer key is desired, it is generated on a file at the same time the test is being typed. It may be printed following the printing of the test.

While this system has substantially improved our capacity to generate high quality tests, it has some drawbacks. The choice of APL as the programming language was an excellent one from the standpoint of programming ease and ease of alteration, but APL is very slow. Translating an expression such as "[x + [1@y]]@[x - [2@y]]" takes about a second of central processor time on a fast computer. It also uses a good deal of memory. The translation of the above 23 character expression turns out to be 349 characters long. If CPU time is cheap and memory is cheap these are not great drawbacks. But if either is expensive, alternatives may have to be found.

If CPU time is expensive, a language which produced more efficient object code might be used to speed translation time. A disadvantage of this alternative would likely be a significant increase in programming time. Also, our system makes use of the fact that APL allows dynamically alterable data types and is recursive. Without these features much of the conceptual simplicity of the present system would be lost, making programs more difficult to use as well as to alter.

3. Express as radicals of the same order and multiply. Write the answer in simplest radical form. $\sqrt{5x} \cdot \sqrt[3]{5}$

4. Write the product as a single fraction in lowest terms:

$$\frac{1-y3}{y+y^2+y^3} \cdot \frac{1-3y-4y^2}{1-y^2}$$

5. Factor completely: $4y^4-6y^3-2y^2$

Figure 3. Sample test questions.

Another possibility would be to do translations of mathematical expressions on a one-shot basis, with the translated versions of the items stored in the item banks. A disadvantage of this is that it increases the second problem: heavy use of memory. One solution would be to keep the item banks on magnetic tape, and only load the relevant ones when generating tests. This is a cumbersome alternative, and not recommended if it is desired that that the system be readily available to users who have limited experience with computers. A better solution might be to take advantage of recent developments in production of low-cost computing hardware, such as microcomputers with floppy disc mass memory. The latter alternative is one we are moving toward in our system.

RECORD MANAGEMENT

The record management component is an interactive record keeping system designed to keep track of the grades for any number of class sections, each containing 200 or fewer students. It assumes a course structure in which students progress through a series of units on which they are tested and, if necessary, retested in order to assure subject matter mastery in small increments.

Three major functions are performed by the use of the system: (1) initialization at the beginning of the course, (2) entering of grades, and (3) retrieval of information from the system.

All programs are written in APL, and execution of a command by a user usually involves simply naming an APL procedure to be performed. Since APL is interactive, this is generally done on-line through a terminal connected to a central computer. Users need only know the syntax for invoking procedures, which is very simple, and how to sign on to the system.

Data and programs are stored in files which may be accessed by the command ")LOAD filename." The data for each class section is contained in its own set of files, but a user needs to access directly only the one file which contains the procedures he needs to invoke. These "bootstrapping procedures" (each of which corresponds to a "command" in the user's lexicon) do all further fi[illegible]ndling and data manipulation. Hence [illegible]e from the need to be able to sign [illegible] load a single file, users need na[illegible]knowledge of the computer.

When a user types a command, what usually happens is that this results in a certain set of programs being loaded from a file, and relevant data being loaded from one or more other files. The programs operate on the data files, and make changes or printing out results, as appropriate. A sample of some of the command procedures follows. Figure 4 contains some illustrations of their use.

Initialization, or setting up files for a class section, is performed by the user, who invokes the procedure INITIALIZE. Relevant data (course, section, number of units, grading scale) are prompted, and appropriate files are set up. A second procedure (NEW) enables the user to enter name and other biographical data for each student.

```
      NEW

 TYPE IN  NAME,COLL,SOC. SEC. NUMBER

 EXAMPLE:   BLOW,JOE LA 123456789

 TYPE  END  TO TERMINATE

 SPADE,SAM BA 333881978
 UP,BOTTOMS LA 222771979
 GLOVES,KID PHA 303050607
 END

       MAKEUP

  RABBIT 2 R 27
  SPADE 1 T 18
  END
  RABBIT,JACK     , UNIT 2: 17 2 27 8
  SPADE TOOK THIS TEST   AND GOT 14
  SPADE,SAM       , UNIT 1: 18 2 31 11


            PI'GLOVES'

     GLOVES,KID
      10/26  2/23  12  35 F
       -/-   -/-

       LOWGRADES 2

 MATH 21,  SECTION 333
 PROJECTED LOW GRADES

 NAME            PTS PRJ GR  COLL  SS NUM

 GLOVES,KID       12  36  D  PHA  303050607
 UP,BOTTOMS        4  12  F  LA   222771979

       DROPLIST

  RABBIT,JACK    FA    20 3 17   TO MATH 6
  GLOVES,KID     PHA   12 3 17   "GOT BEHIND"
```

Figure 4. Sample "commands" in management system. (Circled items were typed by user; others by the computer.)

Grades may be entered as a block (ENTER) or individually (MAKEUP). Since in our courses most students take their tests at one time, the block method is usually used.

Results may be printed out for the whole class (PRINTU) or for individuals (PI). PI returns results for all units completed plus a projected grade for an individual. Others generate high speed printer listings by name or social security number of all results for the entire class. Another (LOWGRADES) prints a listing of students whose progress indicates they will receive a low grade, together with relevant biographical data. Another (DROPLIST) prints a listing of students who have dropped the course, with commentary on why and when dropped.

Many of the drawbacks associated with the test generation component do not occur in the record management component. CPU time is not an important factor, nor is memory capacity. Although there is a continuing need to alter the system as course structures and grading schemes change, the modular structure of the system and the fact that it is written in APL make this relatively easy to do.

DISCUSSION

The system described above reflects a definite philosophy about the development of a record management and test generation system, one not necessarily appropriate to all situations. It represents a commitment to simplify software development and alteration at the cost of increased central processor time and in some cases memory requirements. It is perhaps more appropriate to a relatively small-scale program such as Drake's (approximately 700 students and 6 teachers per year), than to one serving several thousands of students and scores of teachers. In the latter case, CPU time and memory might be a much more important factor.

If there is a message in this project, it is that advances in computing in a number of areas have lessened the need for small-scale programs to depend on the production of materials or systems from outside sources. Among the developments which encourage small-scale projects are the existence and widespread understanding of powerful languages, and the rapidly decreasing cost of computer time and memory. They provide an opportunity not only for schools to develop their own computer-based programs, but to develop them in ways appropriate to their individual needs and philosophies.

HARDWARE

A MICROCOMPUTER SYSTEM
FOR BEHAVIORAL EXPERIMENTS

George Koide and R. Lee Howard
Hilo College
P. O. Box 1357
Hilo, HI 96720

Psychology laboratories which service both teaching and research activities require that the experimental control equipment be versatile, convenient, and reliable. Much of the commercially available card and patch-panel control equipment is cumbersome to program, and requires a surplus of expensive spare items in order to maintain flexibility. In contrast, the microcomputer-based system to be described can provide a low-cost and highly versatile programming unit in which control instructions can be readily modified.

Introduction

Individual departments in many smaller schools and colleges have chosen to purchase "minicomputer" systems as an alternative to the expense and inconvenience of using system computers maintained by the larger institutions. The minis are generally used for real-time data handling for teaching and research laboratories.[1,2,3] However, the cost of even the smaller minicomputers is sometimes prohibitive.

This paper presents an economical microcomputer system, costing less than $2,000, which is being used in our laboratory. The system controls reinforcement contingencies and records and stores data from two independent operant learning experiments. The variable of interest in these particular experiments is response "bursts" related to reinforcement presentation.

Hardware System

The hardware system consists of a Southwest Technical Products microcomputer, three peripheral interface adapters (PIAs), and five input/output devices as shown in Fig. 1. The microcomputer consists of a M6800 microprocessor and 8K of static memory. A 1.7971 MHz crystal generates a timing signal which is used by the microprocessor.

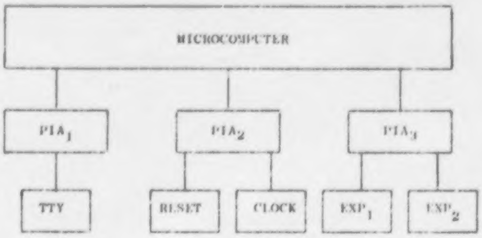


Figure 1. Block diagram of a microcomputer system for two simultaneous real-time experiments.

The first peripheral interface adapter connects a teletype (ASR-33) to the microcomputer. Programs punched on paper tape are read into the microcomputer from the paper tape reader and the results of the experiments are printed out on the print unit.

The microcomputer communicates with the real-time experiments through the other two peripheral interface adapters. An active signal on the first communication line of the second PIA (PIA_2) resets the experiment to the beginning of the program and resets the absolute timer to zero. The second line of PIA_2 is connected to a clock circuitry which converts the 60 Hz power line frequency to TTL compatible 60 HZ pulses to interrupt the computer at regular intervals. Thus the microcomputer keeps time by counting these interrupts. The last two lines connected to PIA_3 are used to collect responses from the two independent real-time experiments. The microswitches from the skinner box are connected to one-shot circuits for interrupting the computer. This PIA is also used to communicate back to the skinner box whenever a reward is made. To activate a reward, a store accumulator instruction is performed to transfer data from the accumulator to the PIA. The PIA is programmed so that a positive pulse is produced on

---

[1] Greeg, Lee W., "Computers: Large-Scale Usage in the Balance," American Psychologist, 30:199 (1975).

[2] Castellan, N. John Jr., "The Modern Minicomputer in Laboratory Automation," American Psychologist, 30:205 (1975).

[3] Bailey, Daniel E. and Peter G. Polson, "Real-Time Computing in Psychology at the University of Colorado," American Psychologist, 30:212 (1975).

its CA(B)2 output line. This pulse then fires a one-shot (½74123) which triggers a Reed relay. The reward mechanism of the skinner box is connected to the Reed relay and a reward is made everytime it is shorted. The pulse of the one-shot must be long enough to activate the feed mechanism of the skinner box.

The clock circuit consists of a filament transformer which reduces the power line voltage to 6.3V rms. This 6.3V rms sinusoidal signal is partially shaped to square pulse by a 6.2V Zener diode as shown in Fig. 2 and its voltage is further reduced by a voltage divider network which consists of two series resistors. This signal then passes through a CMOS buffer and triggers a debounce circuit which fires a one-shot. Finally, the pulse from the one-shot circuitry is used to interrupt the microcomputer.

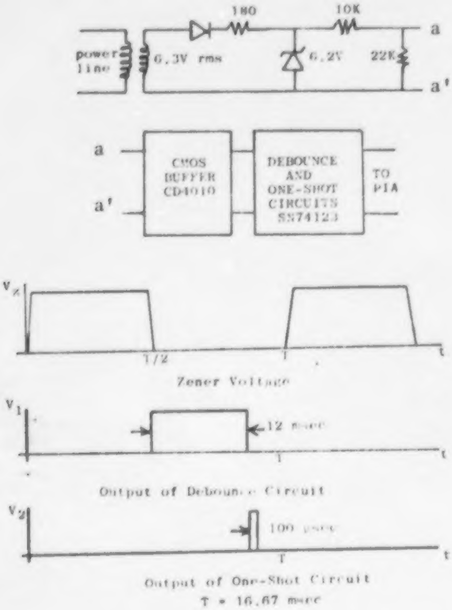

180
10K
a
power line
6.3V rms
6.2V
22K
a'
a
a'
CMOS BUFFER CD4010
DEBOUNCE AND ONE-SHOT CIRCUITS SN74123
TO PIA
V_z
T/2
T
t
Zener Voltage
V_1
12 msec
T
t
Output of Debounce Circuit
V_2
100 μsec
T
t
Output of One-Shot Circuit
T = 16.67 msec

Figure 2. Diagram of clock circuit and timing signals of the 60 Hz signal, debounce circuit, and one-shot circuit.

The debounce and one-shot circuits use the SN74123 dual monostable multivibrator integrated circuit. The debounce circuit produces a 12 msec pulse on a high-to-low transition. This pulse then triggers the one-shot on a negative transition and provides a 100 μsec pulse which is required to interrupt the microcomputer.

Software

For this particular application of controlling two real-time experiments, machine language programming was used in order to save memory space and keep software development cost low. With 8K of memory, we found most of it was needed to store data. Assembler language may be employed if an assembler or cross-assembler is available, however, we felt the purchase of an assembler was not justified since its cost is about a thousand dollars--more than double the cost of the microcomputer. BASIC, which is readily available and economical, may be used if memory space is not a problem.

The program to control the simultaneous real-time experiments and to record and store data is broken into two major sections. See Fig. 3 for the flowchart diagram of the program. The first section initializes the peripheral interface adapters, clears the memory portion which stores the data, and sets the various pointers and counters to their appropriate values. Then the program enters an infinite loop until an interrupt occurs.

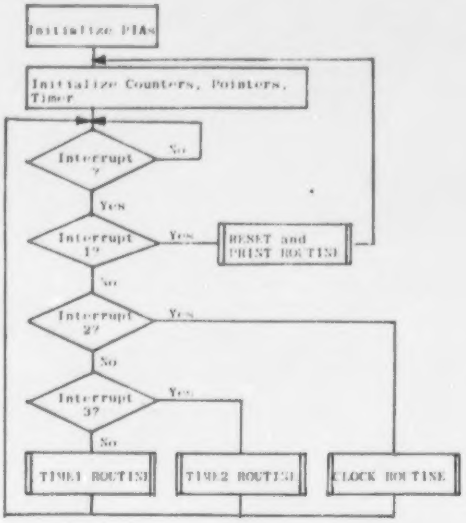

Initialize PIAs
Initialize Counters, Pointers, Timer
Interrupt ?
No
Yes
Interrupt 1?
Yes
RESET and PRINT ROUTINE
No
Interrupt 2?
Yes
No
Interrupt 3?
Yes
No
TIME1 ROUTINE
TIME2 ROUTINE
CLOCK ROUTINE

Figure 3. Flowchart diagram of real time experiment.

The peripheral interface adapters must be programmed before they may be used to interface the microcomputer with the input/output devices. The MIKBUG[4] ROM initializes the control registers of the PIA which interfaces the microcomputer with the teletype whenever the microcomputer is reset. The other peripheral interface adapters are then programmed so that a high-to-low transition from the one-shot circuits of the communication input lines causes an interrupt. These control registers of the PIAs are set at the beginning of the program and are not changed during the execution of the experiments.

The second section consists of an interrupt service routine for the four interrupt lines. The priority of servicing these interrupt lines are arranged in the following order: (1) PIA$_2$ line 1, (2) PIA$_2$ line 2, (3) PIA$_3$ line 1, and (4) PIA$_3$ line 2.

---

[4] MIKBUG is a tradename of Motorola, Inc.

The reset interrupt line is used before and after an experiment. Before an experiment is performed, the timer may be set to zero by using the reset line. During an experiment, the reset line should not be used. After the experiment is completed, the reset line is activated to print out the results.

The reset interrupt line is assigned the highest priority. This portion of the program uses two subroutines (SUB1 and SUB2). Subroutine SUB1 prints out the heading of the experiment. Since most of the headings are identical for both experiments, only minor changes are made within the first interrupt program segment to ensure the correct titles for each experiment.

Subroutine SUB2 performs several functions. First, it converts the absolute time stored in two bytes of memory from binary to BCD. The result of the operation occupies three bytes of memory and is printed out by the teletype print unit (Since time is kept in tenths of a second, an experimental run is limited to about 108 minutes or nearly two hours.). Next, the difference in time of the two adjacent responses is calculated. Since the difference is stored in binary, it is also changed to BCD and printed out. This process is repeated until all data are printed. Finally the subroutine returns control to the first interrupt program segment.

Subroutine SUB2 requires two argument values from the main program. The first value is the number of responses stored in memory, and the second value is a pointer which locates the address of the first time response stored in memory. All other data are stored in sequential order in ascending address locations. Each experiment is alloted 2K of memory for storing data. Thus 1K of responses may be stored since each response requires two bytes of memory.

After the results of both experiments have been printed, the timer, counters, and pointers are reset to their initial values so that both experiments can be repeated.

The clock routine uses two bytes of memory to store the absolute time in tenths of a second. A "divide-by-six" routine is used to change the input clock frequency of 60 Hz to 10 Hz.

The routines, TIME1 and TIME2, are written to service the two experimental interrupt lines of the real time experiments. These routines transfer the time kept by the clock to the appropriate memory cell. A counter stores the number of responses and a pointer is used to place the time in sequential memory cells.

Summary

Many real-time experiments can be economically controlled by a microcomputer. With the PIAs, a microcomputer can easily be made to communicate with various devices. Vendor-provided software is sometimes limited to assembler and BASIC, thus most programs must be written in these languages or in machine language. On the other hand, if the microcomputer is to be used as a dedicated system for a specific application, the program may be stored in a ROM or PROM board and plugged into the microcomputer whenever needed.

For slow processes such as animal experiments, a microcomputer can control several simultaneous experiments and do some simple data analyses. For faster processes, several microcomputers may be used in parallel. In addition, if one or more microcomputers are interfaced with a minicomputer, they are able to relieve the larger system of tedious I/O processing, thus freeing the minicomputer for more complex data analysis or for program development. Since intermediate steps are eliminated, data transfer will be more economical and will be less error prone because of the direct communication links between the computers.

SOFTWARE

DATA STRUCTURES FOR SOPHOMORES

by
John Beidler
John Meinke
Dept. of Math/Comp.Sci.
University of Scranton
Scranton, Pa. 18510

In the art of programming, programmers manipulate two resources, program structures and the structure of their data, to solve problems using various hardware configurations. The tradeoffs between program structures and data structures affect the time and memory requirements of programs and can dramatically effect the resources that are required to solve a problem. Because of its importance, just where does Data Structures belong in an undergraduate curriculum? Because of its relationship to Discrete Structures, should it be taught after Discrete Structures? Curriculum '68 places Data Structures as an upper division coures or possibly even combined upper division/graduate course, which has Discrete Structures as a prerequisite. This suggests a more theoretic and less pragmatic approach. We disagree. In this paper we describe our approach to Data Structures which allows us to make it a sophomore level required course.

Our Data Structures Course, first taught in 1973, was originally conceived as the keystone course in our computer science curriculum. Our curriculum's original design was influenced by Curriculum '68 and the undergraduate computer science program at Penn State University. However, we had to adapt our program to a smaller college environment and include a curriculum track which allows a business emphasis. We felt that regardless of which track in our program a student might follow, theory/systems or business/DP, there must be a strong nucleus common to both tracks. For this reason we concentrated on developing four strong courses that would be given as the freshman and sophomore year component of our curriculum.

As part of this, we also saw the need for software tools to support these courses. Through this, a structured programming preprocessor evolved. However,as part of this preprocessor we included timing and dynamic storage allocation features. Through a strong emphasis on structured programming that begins with our first computer science course, our second course inteoduces many discrete structure concepts-queues, stacks, trees, graphs, etc.. Our third course is a course in assembler level programming and computer organization. This provides the foundation for our course in Data Structures.

Since the basic structures are introduced in the second course, we feel that our Data Structures course has two principal goals. The first is to present a great variety of techniques for representing the various structures. This includes dynamic storage allocation and management techniques. The second goal is measurement. The students learn to measure the time and space costs of their solutions to problems as well as the time and space costs of their data structure techniques. We believe the measurement aspect of the course is extremely vital in that we do not just use sophisticated data structure techniques for their own sake.

During the past two years, this course has been greatly improved by the use of our preprocessor, PLUS, and its data structure capabilities. The data structures in PLUS fall between the sophisticated abilities of PASCAL and the GETMAIN/PUTMAIN of PL/1. The RECORD statement in PLUS establishes a template for the records that will be dynamically allocated. GETRECs establish pointers to records which are "created" as they are needed and PUTRECs return records to the available space. RECORDs can be allocated and linked as desired. Information is accessed from records through a pointer/displacement access method where the pointer indicates the record that is to be accessed and the displacement indicates the position that is to be accessed.

Our success with this course can be measured in two ways. First, our original intention was: with this course in their background our students could be required to do more sophisticated assignments in upper level courses and hence, in turn, our upper division courses would be improved. This has been the case and we have been pleased with this result.

Our second measure of success has been the number of our graduates who have told us of their successes and expressed the feeling that a key element in their background which got them jobs or helped them tackle difficult tasks has been the data structures course.

Introduction

In the art of programming, programmers manipulate two resources, program structures and the structure of data, to solve problems using various hardware resources. The trade-offs between program structures and data structures effect the time and memory requirements of programs and can dramatically effect the resources that are required to solve problems. Because of their importance, data structure techniques should be taught early in a computer science curriculum. However, some of these topics appear to need some sophisticated mathematical prerequisites. This would seem to indicate that many data structure topics must be delayed until the junior or senior year. With these apparent contradictions, just where does a Sata Structures course belong in a curriculum?

Although our curriculum was influenced by ACM's Curriculum '68, it had to be tempered by our school's faculty and student body, as well as our computing and financial resources. Within these constraints we have developed a broad based computer science curriculum which prepares students for a variety of computer related job opportunities as well as graduate school possibilities in several areas. In order to provide this broad set of opportunities, it is necessary to provide a strong nucleus which both prepares the students for their various options as well as assisting them in selecting an option which best meets their interests and abilities early in the curriculum.

It wasour conclusion that a strong Data Structures course could be offered in the sophomore year. Many factors went into this decision. Specifically, we believe this course is the keystone of our curriculum. A discussion of the factors involved in this decision appears in the second section of this paper.

The Data Structures course has been taught every year since 1972. Interestingly, this course has used a different text or set of course notes each year it has been taught. As this course evolved, it has been influenced by current trends in computer science. Perhaps the greatest influence on this course has been the collection of topics which are loosely joined together under the title of Structured Programming. To support a greater emphasis on problem solving while down playing programming language idiosyncracies, a collection of software tools was brought together in a FORTRAN preprocessor we call PLUS(Programming Language Using Structure). PLUS is the major software tool/programming language used by our students. In 1976, a data structure capability was added to PLUS. This data structure capability handles many of the implementational details of dynamic storage allocation while allowing the user to concentrate on the formation and manipulation of his structures. PLUS and its data structure capabilities are described in greater detail in section three.

With this software tool, students have been able to successfully handle a large number of complex programming assignments. A natural result of this was to redesign the Data Structures Course to take greater advantage of the data structure capabilities and other software tools in PLUS. This allows us to spend more time on the measurement of the time and space requirements of algorithms. A discussion of the course syllabus appears in section four.

Curriculum Overview

Our original decisions on the design of our curriculum were based on ACM's Curriculum '68 and was modified by our needs and resources. We believe our original curriculum structure has been proven successful by the job placement and graduate school opportunities offered to our students. Also, we believe other curriculum recommendations, in particular, the recommendations for small schools, by Austing and Engel, the new ACM C^3S recommendations, and the curriculum recommendations of the IEEE prove the validity of our curriculum structure.

Our curriculum is based on a nucleus of four computer science courses which form the freshman and sophomore components of the curriculum. The freshman year course sequence emphasizes problem solving and structured programming from the beginning. This approach with the use of our structured programming preprocessor, PLUS, places less emphasis on learning all the idiosyncracies of a programming language and places greater emphasis on problem analysis and solution design.

This approach is continued in the second course and in the Assembler course which is given in the first semester of the sophomore year. These two courses serve to broaden the student's awareness of the breadth of computer science. Besides seeing programming at several levels, the students are also exposed to some of the basics of data structures; queues, trees, stacks, etc.. However, at this level, the emphasis is on the particular topic under discussion and the data structure is presented only as a convenience for solving the problem. Even then, the emphasis is not on the data structure, but their representations.

At this point we feel that Data Structures should be presented. Since it is a sophomore level course, we feel it is best that the course be pragmatic in nature. By presenting it at this point in the curriculum, we believe the students are well prepared with a broad collection of structuring concepts and a large variety of ways to carry out these concepts. One drawback that might be expected with this sophomore level presentation is that it uses some discrete structure concepts. We believe, however, that beyond the basic concepts, these two courses head in two different directions. Discrete structures parallels in approach and content a mathematics course in Modern Algebra. On the other hand, Data Structures is concerned with methods of representing these structures. For these reasons, rather than having Discrete Structures as a prerequisite, a simple presentation of the necessary concepts is part of the Data Structures course.

We have other reasons for wishing to present this course at the sophomore level. One important reason is that we wish to present a course which gives students an excellent idea of computer science and relieve them of any misconceptions they might have about computer science. At this level, students who still have doubts about computer science as their major can select and change to other majors without much difficulty.

A third reason for this sophomore level presentation of Data Structures is our feeling that it strengthens our upper division courses. Specifically, we feel that with the course at this level, other computer science courses can take advantage of this and require more advanced programming assignments. Further, this should be expected regardless of whether the courses are theory oriented, business oriented, or systems oriented. In this regard, we feel that this approach has been highly successful.

A Software Tool for Data Structures

A key element in the development of our Data Structures course has been the structured programming preprocessor, PLUS, and its software tools. Among these tools is a capability which supports dynamic storage allocation. This feature allows the user to concentrate on the creation and linkage of his structures while the preprocessor takes care of the implementational details. Obviously, there are more sophisticated data structure capabilities. However, this capability fits within a FORTRAN framework which gives very immediate and general portability. General details on the preprocessor appear in a previous paper.

Appendix I contains the syntax of the RECORD statement. The RECORD statement describes the layout of dynamic records and the names of pointers that can access the various types of records. Records are created by the GETREC function and erased by the PUTREC subprogram as in

```
      point=GETREC(name)
```

and

```
      CALL PUTREC(point, name)
```

where "point" is a pointer and "name" is the type or record being created and erased. Positions within records are accessed by a pointer-displacement access method. For example, of a list of records were created where the records are defined by the statement

```
RECORD POLY(LINK/P,COEF/R,EXPON/I)-
   NEW-POINT
```

then each GETREC(LIST)would create a record consisting of three positions. The first position in the record is used as a pointer (to other records of this type or is null), the second position can contain a real value, and the third is a integer value. If a pointer to a record is established,

```
  NEW=GETREC(POLY),
```

then any position within the record can be accessed and used in a

```
  point @ displacement
```

format.

This is illustrated in example 1 with a subroutine which adds polynomials which are dept as ground link lists. Each record in each of the lists pointed to by A,B, and SUM is a three position record and the appropriate positions in each record are accessed by having the combination of a pointer to a record along with a displacement into the record. The @ operation has the highest precedence and if next to each other, they are decoded from left to right. Also, parentheses cannot be used to alter this. For example,

```
X=POINT@LINK@COEF
```

would obtain the COEF item from the record pointed to by the LINK of the record the POINT indicates. If that sounds confusing simply stated, if POINT had pointed to the front of the list, then the COEF from the second record in the list would be copied into X.

There are no limitations on the number of pointers that can be in each record. For this reason, other more complex structures can be created and manipulated using this approach. Example two illustrates a routine which does a natural order tree search through a tree and calls another routine each time it finds a "marked" record. This routine is actually part of PLUS and is used to produce a cross reference listing. Each time the PRTWRD routine is referenced, it can use the PARENT pointer to form and print a word. The "marked" displacement is actually a pointer to a list of the lines where the word is referenced.

To enhance the data structure capabilities, several additional features have been included. One feature is a variable displacement capability. This was found useful in clarifying several tree manipulation algorithms, including the algorithm which builds the dictionary tree used to form the PLUS cross reference listings.

The RECORD statement and record access support one, two, and thre dimensional arrays as pointers and as displacements within records. For example, the records created through reference to

```
RECORD TREE(CHILD/P,SIB/P,BOARD(3,3,3))
 -POINT(3)
```

could be accessed by

```
POINT(I)@BOARD(J,K,L).
```

Our experiences of the past two years have been most encouraging. This data structure capability has been used by students in at least five other courses. These uses were the student's choices, and they were not forced to take this approach. This capability was also used by GTE Laboratories to create a pseudocode processor which is being used to control, design, and document large programming projects.

Course Structure

Naturally, since we have an excellent software tool for data structures, the course was designed to make maximum use of it. However, we are aware of the danger of overemphasizing this tool and we do not wish this means of learning data structures to become a goal in itself. In order to take advantage of the data structure feature in PLUS, it is desirable to introduce early structures whose representations lend themselves to dynamic allocation, but this is tempered by:

1. A strong emphasis on time and space measurement which we believe to be the most important component in the course;
2. A desire to distinguish structures from their various representations.

Appendix III contains a course outline which describes the order of the topics. After a brief introduction to the control structures of PLUS, we set the framework of the course by looking at several computer and non-computer examples of logical concepts versus their physical representations. This is coupled with some simple time and space measurements of several algorithms. Program measurements is a difficult topic. In order to assist the students the concepts of dominance and order of magnitude are introduced and in program measurement. Students are told to consider six possible time measurements. These are the best, worst, and average number of loop iterations for the worst loop and for total sum of all loops. For some programming assignments, to assist the students, we suggest certain timings be considered. For example, for the quicksort assignment, the students are told to consider two cases of timing the worst loop. These are, determine the time if the numbers are already sorted and time the algorithm if the part of the array being sorted is always bisected. We believe one real value in emphasizing measurement is that students will be cautious about appraising each method against the problem being solved.

Next, we present a brief survey of the discrete structures that are used in this class. In this presentation, the structures are presented without any discussion of how to represent them. This presentation serves as a forum for a discussion of several potential programming assignments. This serves serveal purposes:

1. It generates interest in the course content;
2. It separates the structures from potential representations;
3. It gives the students a chance to observe the value of various structures without concern for how the structure is carried out.

The next two chapters, besides

presenting several simple representations of queues and pushdown stacks set the stage for using the PLUS RECORD feature. The material on queues and stacks ends with an emphasis on the value of pointers and the importance of storage management. Strings and lists illustrate that the desire for a logically sequential structure is not always met best with a physically sequential represention. Here several algorithms for list manipulation are presented. These algorithms anticipate the RECORD statement in PLUS

With the RECORD statement and the use of pointer variables, a variety of list representations are described and presented. From this point in the course we attempt to interleave the use of dynamic RECORDs with other methods or representing structures in order not to make the RECORD capability an end in itself. At this point, when we present RECORDs, no attempt is made to describe how the RECORD statement and RECORD features function. Nor is this information relevant at this point in the course. It would be an unnecessary tangent. This material is placed at the end of the course as part of the discussion on storage allocation techniques.

Trees are presented with several record and non-record representations. A considerable amount of time is spent with trees and tree manipulations. We consider this an important intermediate goal, or natural plateau, of the course. Much of the material that precedes the discussion of trees is used here, queues, stacks, lists, and both RECORD and non-RECORD representations of trees. On the other hand, trees are an invaluable aid in working with some of the more complex structures that follow. Not only do students see good non-RECORD representations, such as uniform binary trees in arrays, but they also oftain a good solid foundation in the manipulation and use of pointer variables.

At this point in the course outline, the cyclic nature of the presentation is apparent as we make a third presentation of the data structure feature in PLUS. What is not so apparent is the cycling of other material. For example, throughout the semester, many sort procedures are used as examples or assignments, then towards the end of the semester they are collected together and their relative merits are discussed.

With this organization, the material on graphs and digraphs is presented about two-thirds of the way through the semester. This gives students sufficient time to carry out at least one relatively complex graph assignment like the traveling salesman problem, a four coloring, or one of many other complex problems. Again, as with trees, both RECORD and non-RECORD representations are described. Here, the continued emphasis on program measurement pays off. Besides comparing the space requirements for both RECORD and non-RECORD representations, these representations are timed. This leads us to an improved RECORD presentation which makes heavy use of pointer variables. This really seems to drive home to the students the value of pointer variables.

The pace of the course is now reduced in order to provide the students with enough time to comprehend the new material that is being discussed in class and complete several programming assignments. The sorting, searching, and mathematical material is fairly standard material. However, with the sorting and searching we begin to look at secondary storage devices and the problems of handling extremely large amounts of data.

The section of the material on storage allocation is approached from the point of view as to how the GETREC and PUTREC routines could function. This also leads into the use of secondary storage with descriptions of the concepts of paging and virtual memory.

The last section of material is here representing one natural next steps, namely, applying the concepts to very large collections of data and the design philosophies of information management systems. This is a cursory presentation whose purpose is simply to make students aware, in general, of various general data base approaches and their general design.

During the semester, students can select from about twenty assignments. The better students will complete about twelve to fifteen assignments. To pass the course a student must complete at least eight but normally ten assignments. Of these;

1. At least three must use RECORDs;
2. One must use RECORDs to represent trees;
3. At least four assignments must contain proper timings;
4. One assignment must utilize random number generator to thoroughly test the program;
5. One must be an advanced sort technique.

Conclusions

We believe comments on three items are appropriate, PLUS as a software tool, the course content and structure, and the position of the course in the curriculum. With regard to PLUS, we were concerned with whether PLUS is a valid software tool or an academic toy. Since its initial development, it has been used in several non-academic environments and on a variety of systems. It has been a user developed tool and as such its capabilities continue to expand. Its data structure capability has been used to develop a complex software documentation processor for GTE Laboratories. It has also been successfully used by several industrial and commercial groups. Its success outside of academia reinforces our trust in it.

The course content is standard but its organization is critical, giving students sufficient time to handle several interesting and complex programming projects. Finally, the position of this course in the curriculum is extremely important. It gives our students an early indication of their sofrware design capabilities and with this course at the sophomore level, upper level undergraduate courses can take advantage of this and build upon it.

Bibliography

1. Austing, R., et.al., C^3S Curriculum Recommendation, SIGCSE Bulletin, v.9,no.2(June 1977).
2. Beidler, J., and Meinke, J., "PLUS- An Approach to Structured Programming with Program Analysis and Documentation," Proceedings of the Computer Science and Engineering Curriculum Workshop, Williamsburg, VA, June 6-7, 1977,pp.83-87.
3. Berztiss, A., Data Structures-Theory and Practice, Academic Press, 1971.
4. IEEE Computer Society, A Curriculum in Computer Science and Engineering, 1977.
5. Pfaltz, J., Computer Data Structures, McGraw-Hill, 1977

APPENDIX 1

SYNIAX OF RECORD STATEMENTS

```
<REC STMT>::=RECORD <REC LIST>
<REC LIST>::=<REC DESC>
               !<REC DESC><REC LIST>
<REC LIST>::=<TITLE>(<DISP LIST>)-<POINT LIST>
<TITLE>::=<VFI>
<DISP LIST>::=<LAST DISP>!<DISP NAME>,<DISP LIST>
<POINT LIST>::=<VFI>!(<VAR DISP LIST>)
               !<VFI>,<POINT LIST>
<LAST DISP>::=<DISP NAME>!<VARIABLE NAME>()
<DISP NAME>::=<VFI>!<VFI>/<TYPE>
<VAR DISP LIST>::=<NON-INDEX VARIABLE>
<TYPE>::=I!P!R!L!P <TITLE REF>
<TITLE REF>::= '<TITLE>' OF A RECORD
<VFI>::='VALID FORTRAN INDENTIFIER(1,2,OR 3 DIM.)'
```

SYNTAX OF RECORD REFERENCES

```
<REC REF>::=<POINT>@<DISP>
<DISP>::=<VFI>!VFI@<DISP>
<POINT>::=<VFI>
```

SEMANTIC INTERPRETATION

FOR EACH @, THE <VFI> PRECEDING IT MUST BE A POINTER TO A RECORD AND THE <DISP> MUST BE A DISPLACEMENT IN THAT RECORD.

*

Appendix 2

1. Preliminaries
   Introduction to PLUS
   Using a Random Number Generator
2. Basic Philosophy of Course
   Structure-Logical vs Physical
   Trade-offs-time and space measurements
   Timing Techniques
3. Logical Structures (Discrete Structures)
   Queues, Stacks, Strings, Lists
   Graphs, Diagraphs, Trees
4. Queue and Stacks (array representations)
   Basic representations
   Multiple structures in one array
5. Strings and Lists
   Logical sequential vs physical sequential Introduction of general pointer concept
6. Pointers and Dynamic Records
   Record descriptors and recordacess
   Examples
7. Lists
   Grounded, Two-eay, circular
8. Trees
   Binary-array representation
   General-using child and sibling pointers
   Tree searching methods
   Tree access using a parent pointer
9. Additional Record features in PLUS
   Describe additional capabilities
10. Graphs
    Matrix Representations and Triangular Form
    Dynamic record representation
    Improved dynamic approach
11. Sorting Methods and their evaluations about 10 sort methods described.
12. Searching
    Sequential, Binary, Indexed Sequential,Hashing,AVL tree
13. Odds and Ends
    A variety of examples
14. Storage allocation Methods
    Various methods and garbage collection
15. Introduction to Data Base Management
    Briefing history and philosophy of standard models.

```
SUBRØUTINE NØTS(TREE)
RECØRD TREERD(PARENT/P,CHILD/P,SIBLING/P,CHAR/I,MARK/I)-TREE-PØINT
PØINT=TREE
DØ UNTIL (PØINT.EQ.0)
:  IF(PØINT@MARK.NE.0)THEN
:      ....
:      ....  INSERT CØDE HERE TØ MANIPULATE NØDE
:      ....
:  ENDIF
:  IF(PØINT@CHILD.NE.0)THEN
:      PØINT=PØINT@CHILD
:  ELSE
:      DØ UNTIL(PØINT@SIBLING.NE.0)
:      :  PØINT=PØINT@PARENT
:      ENDDØ
:      PØINT=PØINT@SIBLING
:  ENDIF
ENDDØ
RETURN
END
```

FIGURE I

```
SUBRØUTINE PØLYAD(SUM,A,B)
INTEGER GETREC
RECØRD PØLY(LINK/P,CØEF/R,EXPØN/I)-SUM-A-B-APNT-BPNT-NEW
APNT=A
BPNT=B
SUM=0
DØ UNTIL(APNT.EQ.0.AND.BPNT.EQ.0)
:  IF(APNT.EQ.0.ØR.(BPNT.NE.0.AND.BPNT@EXPØN.GT.APNT@EXPØN))THEN
:      NEW=GETREC(PØLY)
:      NEW@EXPØN=BPNT@EXPØN
:      NEW@CØEF=BPNT@CØEF
:      NEW@LINK=SUM
:      SUM=NEW
:      BPNT=BPNT@LINK
:  ELSEIF(BPNT.EQ.0.ØR.APNT@EXPØN.GT.BPNT@EXPØN)THEN
:      NEW=GETREC(PØLY)
:      NEW@EXPØN=APNT@EXPØN
:      NEW@CØEF=APNT@CØEF
:      NEW@LINK=SUM
:      SUM=NEW
:      APNT=APNT@LINK
:  ELSEIF(APNT@CØEF+BPNT@CØEF.NE.0.0)THEN
:      NEW=GETREC(PØLY)
:      NEW@EXPØN=APNT@EXPØN
:      NEW@CØEF=APNT@CØEF+BPNT@CØEF
:      NEW@LINK=SUM
:      SUM=NEW
:      APNT=APNT@LINK
:      BPNT=BPNT@LINK
:  ELSE
:      APNT=APNT@LINK
:      BPNT=BPNT@LINK
:  ENDIF
ENDDØ
CALL TSIL(SUM)
....
....  TSIL IS A PRØCEDURE THAT REVERSES THE LINKS
....
RETURN
END
```

FIGURE II

STRUCTURED DESIGN, STRUCTURED PROGRAMMING
STRUCTURED WALKTHROUGHS
AND
THE NEW PHILOSOPHY OF COMPUTER PROGRAMMING

THOMAS J. CASHMAN
LONG BEACH CITY COLLEGE
LONG BEACH, CA

For the past two decades, the software industry has been plagued with difficulties in the development of software that has been unreliable and difficult to modify and maintain. Research by computer scientists has indicated that structured programming could offer a solution to these problems. In recent years, emphasis has been placed upon the principles of structured design in combination with techniques of structured programming to assist in software development. In addition, structured walkthroughs, the peer reviews of program design and coding, appear to offer great potential for improving the quality of software produced. The use of structured design, programming, and walkthroughs in an academic environment can result in a new philosophy in teaching computer programming, resulting in the production of software in which it is expected that programs will execute properly the first time placed on the computer.

INTRODUCTION

The data processing industry and the stored programming concept is about to enter its fourth decade of progress. Beginning with the first electronic computer and the stored programming concept over 30 years ago, the industry has experienced revolutionary growth both in hardware and software. The data processing industry, however, has been plagued with a number of continuing problems. These problems include (1) Programmers have been relatively unproductive, (2) Software developed has often been unreliable, (3) Software developed has often been difficult to modify and maintain. These and other problems have been identified by various authorities, but it has been only during the past decade that the industry has seriously attempted to identify and study the problems associated with the creation and maintenance of software.

Research by computer scientists during the past decade indicated that structured programming could be of value in solving some of the basic problems in software development. In the early 1970's, much of the discussion relating to structured programming centered around the merits of GO-TO-LESS programming. In COBOL, early efforts in structured programming were often characterized by the absence of GO TO statements, merely because GO TO's were not to be used, and by the excessive use of the PERFORM verb to achieve structure. It soon became apparent that structured programs could be just as poorly written as those written in an unstructured form; that is, programs written without the use of GO TO's were not necessarily "good" programs as the programs were not necessarily easy to modify and maintain and were not always reliable.

In recent years, increasingly great emphasis has been placed upon "structured design" as the basis for writing structured programs. Stevens, Myers, and Constantine in their article, "Structured Design," appearing in the IBM SYSTEMS JOURNAL in 1974, defined structured design as "a set of proposed design considerations and techniques for making coding, debugging, and modification easier, faster, and less expensive (7) . This article was to have great impact on structured programming methodology. It soon became apparent that structured design provided the basis for good structured programming, and it is now generally recognized that structured design must precede structured programming in order to produce good "structured code."

Unfortunately, much of the research efforts in structured design and structured programming has been on the theoretical level, and few examples have been presented as to how the theory of structured design and related structured programming methodologies can be utilized to produce good programs. The purpose of this paper is to illustrate a practical structured design methodology, utilizing top down development and functional decomposition, that will lead to the production of "good" structured programs in both industry and the classroom. In addition, the implementation of structured walkthroughs in the academic environment as a part of improved programming techniques will be discussed which results in a "new philosophy" in the teaching of computer programming.

INTRODUCTION TO STRUCTURED DESIGN

Historically, when writing a program, the programmer began by reviewing the programming specifications, and then set out to define the detailed steps which were to be taken in order to obtain the solution to the problem. These detailed steps were often expressed by a program flowchart; or in many cases, the programmer began by writing the actual source code. Using this method, the logic for each new situtation was developed as it was encountered within the sequential development of the solution to the problem.

Quite unlike the "old" method of problem solving, the concepts of structured design are based upon the idea of top-down development and functional decomposition, in which the programmer is first concerned with the output that is to be produced, the input required, and the major processing tasks that are necessary to convert the input to the output. The major processing tasks are then further decomposed into functional modules and successively refined until the structure of the program has been designed. After the structure of the program has been defined, the logic for each of the separate modules is developed. Only after the structure of the program and the logic for each of the separate modules has been defined and approved is the coding of the program begun.

This concept can best be illustrated by taking an example and tracing through the basic design process step by step, emerging with the application coded in the COBOL language. In the example application, an Accounts Receivable Register is to be produced. Input is to consist of a series of input records containing the Customer Number, the Customer Name, and the Amount of Purchase. In preparation of the report, if the amount of the purchase is $100.00 or more, a 10% discount is given. If the amount of purchase is less than $100.00, a 7% discount is given. The report is to contain the Customer Number, the Customer Name, the Amount Purchased, the Discount Percent, the Discount Amount, and the Net Amount. Final Totals are to be taken of the Amount Purchased, the Discount Amount, and the Net Amount. The format of the input and output are illustrated in Figure 1 on the following page.

THE INPUT/PROCESSING/OUTPUT CHART

The first step in the structured design process is to complete an Input/Processing/Output Chart. The IPO Chart is the tool used for defining the structure of the program. On the IPO Chart, the programmer specifies the following:

1. The output which is to be produced
2. The input to the program
3. The major processing tasks that are necessary to convert the input to the output

The entries for the sample program are illustrated in Figure 1. Note that the IPO Chart contains the output that is to be produced from the program (Accounts Receivable Register), the input to the program (Accounts Receivable Input File), and the major processing tasks that are necessary to convert the input to the output. These major processing tasks are

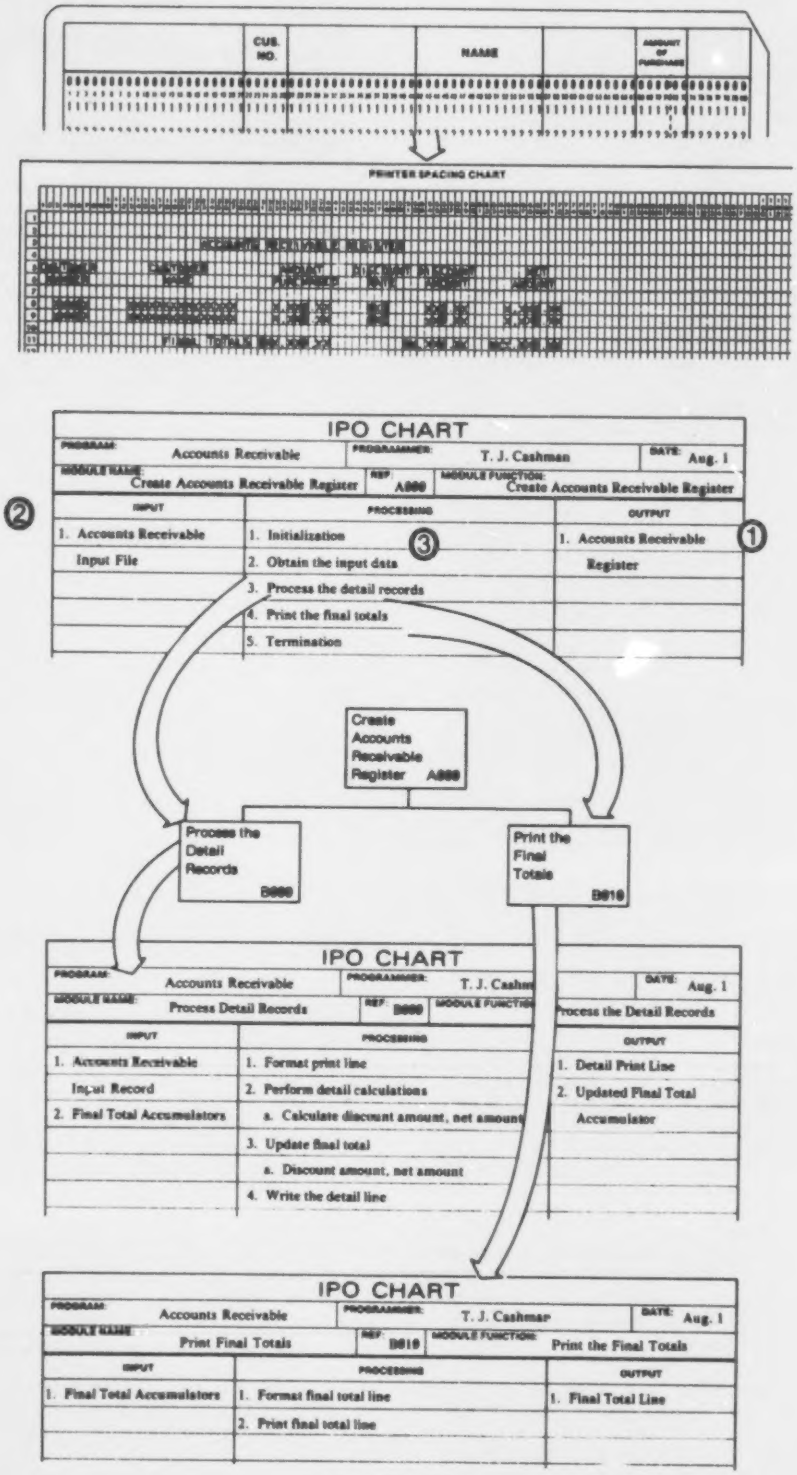
1. Initialization
2. Obtain the Input Data
3. Process the Detail Records
4. Print the Final Totals
5. Termination

In COBOL, initialization involves opening the files. Obtain the Input Data involves the reading of the input records. Processing the Detail Records refers to steps necessary to format the report and perform the necessary calculations. Printing the Final Total involves the formating and printing of the totals. Termination refers to the process of closing the files.

It should be noted that the major processing tasks specified in the Processing Section of the IPO Chart do not specify the manner in which any of the processing is to be accomplished, nor are the tasks necessarily listed in the sequence in which the processing is to take place within the program. Instead, the IPO Chart is used merely to specify the major processing tasks which must occur within the program. Thus, the IPO Chart is used to specify WHAT must take place within the program, not WHEN or HOW.

After the major processing tasks have been defined on the IPO Chart, the programmer should analyze these tasks to determine if any of the major processing tasks requires further decomposition; that is, does it appear that any of the major processing tasks consist of identifiable functions that are of such size or complexity that they should be broken down into separate modules.

In the example, it can be seen that it has been determined that the "Processing the Detail Records" would appear to be sufficiently complex to justify a separate module, and the task of "Print the Final Totals" is a well-defined function that will be considered as a separate module. The relationship of these modules is then illustrated through the use of a


CUS. NO.
NAME
AMOUNT OF PURCHASE
PRINTER SPACING CHART
ACCOUNTS RECEIVABLE REGISTER
FINAL TOTALS
②
③
①
Create Accounts Receivable Register A000
Process the Detail Records B000
Print the Final Totals B010

IPO CHART

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashman	DATE: Aug. 1
MODULE NAME: Create Accounts Receivable Register	REF: A000	MODULE FUNCTION: Create Accounts Receivable Register

INPUT	PROCESSING	OUTPUT
1. Accounts Receivable Input File	1. Initialization 2. Obtain the input data 3. Process the detail records 4. Print the final totals 5. Termination	1. Accounts Receivable Register

IPO CHART

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashm	DATE: Aug. 1
MODULE NAME: Process Detail Records	REF: B000	MODULE FUNCTION: Process the Detail Records

INPUT	PROCESSING	OUTPUT
1. Accounts Receivable Input Record 2. Final Total Accumulators	1. Format print line 2. Perform detail calculations a. Calculate discount amount, net amount 3. Update final total a. Discount amount, net amount 4. Write the detail line	1. Detail Print Line 2. Updated Final Total Accumulator

IPO CHART

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashman	DATE: Aug. 1
MODULE NAME: Print Final Totals	REF: B010	MODULE FUNCTION: Print the Final Totals

INPUT	PROCESSING	OUTPUT
1. Final Total Accumulators	1. Format final total line 2. Print final total line	1. Final Total Line

FIGURE 1

Structure Chart. Thus through the Structure Chart, the overall structure of the program can be defined. From the example in Figure 1, it can be seen that the program is to consist of a module called "Create Accounts Receivable Register," a module called "Process the Detail Records, and a module called "Print the Final Totals."

The next step in the design process is to develop an IPO Chart for each of the lower level modules; that is, the output of each module should be defined, the input determined, and the major processing tasks necessary to convert the input to the output specified. Each of the major processing tasks are then analyzed to determine if further decomposition is necessary. In the example, neither the Process the Detail Records module or the Print Final Totals Module requires further decomposition; therefore, the overall structure of the program has been fully defined.

PSEUDOCODE SPECIFICATIONS

After the programmer has determined the structure of the program and specified what is to take place in the program through the IPO Chart, the programmer must then design the logic within each module. The methodology used to develop program logic is a tool called "pseudocode." Pseudocode allows the programmer to express thoughts in a form that uses regular English language prose and allows for the expression of the flow of control in a program in a straight-forward, easy to understand, manner. Pseudocode has few syntactical rules and merely uses English language phrases to express thoughts, with each phrase representing a programming statement. Psuedocode is written at a level of completeness and detail which allows the coding of the program directly from it.

It should be noted that the pseudocode for each module is developed from the major processing tasks which are specified in the processing portion of the IPO Chart. Thus, the object of pseudocode is to specify the logic and methods of implementing the major processing tasks which are required to accomplish the functions of the module. It will be recalled that the major processing tasks specified on the IPO Chart should indicate WHAT is to occur; while the processing specified on the pseudocode indicates WHEN and HOW these tasks will be performed.

The pseudocode for the sample program is recorded on the Pseudocode Specifications illustrated in Figure 2. As can be seen, the pseudocode is merely a series of statements which indicate the sequence in which operations are to take place within the program and also under what conditions they will be done. It can be observed by reviewing the pseudocode on the following page that GO TO statements are not needed or missed, and similarly, a flowchart to express the logic of the program is not needed or missed.

CODING THE PROGRAM

After the design of the program has been completed and reviewed, the programmer would normally begin the coding of the program. It is important that the same care which is taken in the design of the program be used when coding the program, as the primary source of information for a programmer who is to maintain the program at some later time will be the source listing; therefore, standards of indentation, selection of meaningful data names, and other good programming practices should be followed. The source listing for the top level module is also illustrated in Figure 2.

STRUCTURED WALKTHROUGHS

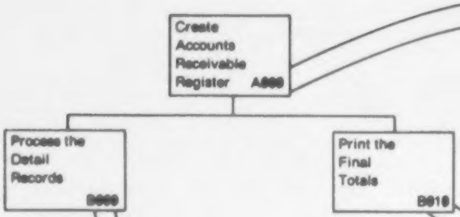
In addition to utilizing the principles of structured design and structured programming to improve programmer productivity, structured walkthroughs have been widely advocated as a means of improving the quality of software developed. The purpose of the structured walkthrough is to have a "peer" review of program design and program coding so that any errors which have been made will be detected in the early stages of the program development cycle, rather than in the midst of final program testing where the corrections are more difficult and costly to make.

The prospect of implementing team concepts and structured walkthroughs in the classroom has been raised by several authorities, most notable of which is Dr. Gerald Weinberg, the author of the book, THE PSYCHOLOGY OF COMPUTER PROGRAMMING. Dr. Weinberg has also advocated the use of the walkthroughs at all levels of programming instruction.

In conducting a walkthrough in an academic environment, the following procedures should be followed:

1. For each programming assignment, the class should be divided into teams of 3 students per team. Upon completion of the program design, the team reviews individually each student's program design. In this session, errors in logic are pointed out, approaches to the solution of the problem discussed, and the design generally reviewed.

2. After it has been agreed upon by the team that the first team member's design reflects a proper solution to


Create Accounts Receivable Register A000
Process the Detail Records B000
Print the Final Totals B010

PSEUDOCODE SPECIFICATIONS

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashman	DATE: Aug. 1
MODULE NAME: Create Accounts Receivable Register	REF: A000	MODULE FUNCTION: Create Accounts Receivable Register

PSEUDOCODE / REF:

```
Open the files
Write the heading lines
Set space control for double spacing
Read an input record
IF there is a record
   PERFORM UNTIL there are no more records
      Process detail records                        B000
      Read an input record
   ENDPERFORM
   Print the final totals                           B010
ENDIF
Close the files
Stop run
```

FILES, RECORDS, FIELDS REQUIRED

Accounts receivable input file
Accounts receivable report file
Heading lines
Printer spacing field
Double space control character
Input record area
  Customer number
  Name
  Amount of purchase
There is a record indicator
No more records indicator

PSEUDOCODE SPECIFICATIONS

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashman	DATE: Aug. 1
MODULE NAME: Process Detail Records	REF: B000	MODULE FUNCTION: Process the Detail Records

PSEUDOCODE / REF:

```
Clear the printer line
Move customer number, name, purchased amount
      to output area
IF purchased amount is less than the amount for
      maximum discount
   Calculate discount amount = amount purchased x
      minimum discount rate
   Move minimum discount rate to output area
ELSE
   Calculate discount amount = purchased amount x
      maximum discount rate
   Move maximum discount rate to output area
ENDIF
Move discount amount to output area
Calculate net amount = amount purchased -
      discount amount
Move net amount to output area
Add purchased amount, discount amount, net amount
      to final total accumulators
Write a line on the report
Set space control for single spacing
```

FILES, RECORDS, FIELDS REQUIRED

Input record area
  Customer number
  Name
  Purchased amount
Printer output area
  Customer number
  Customer name
  Purchased amount
  Discount rate
  Discount amount
  New amount
Amount for maximum discount
Minimum discount rate
Purchased amount accumulator
Discount amount accumulator
Net amount accumulator
Printer spacing field
Single space control character

PSEUDOCODE SPECIFICATIONS

PROGRAM: Accounts Receivable	PROGRAMMER: T. J. Cashman	DATE: Aug. 1
MODULE NAME: Print Final Totals	REF: B010	MODULE FUNCTION: Print the Final Totals

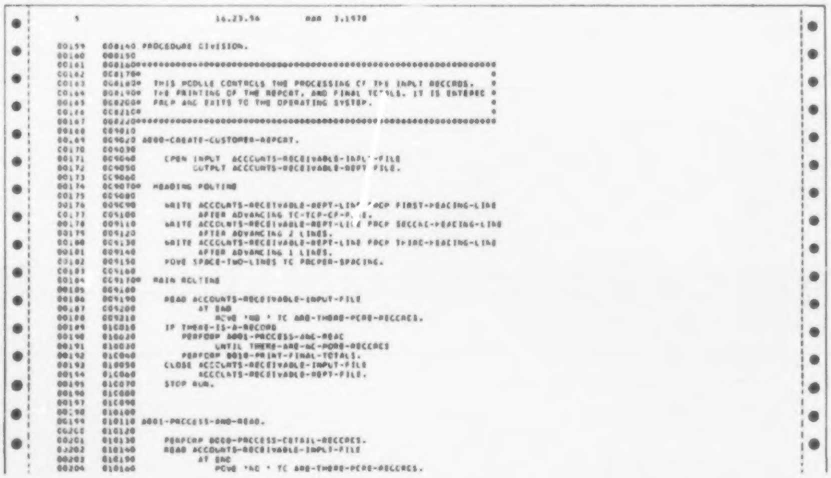
PSEUDOCODE / REF:

```
Move amount purchased to final total line
Move discount amount to final total line
Move net amount to final total line
Write the final total line on the report
```

FILES, RECORDS, FIELDS REQUIRED

Final total line
  Amount purchased
  Discount amount
  Total amount
Amount purchased accumulator
Discount amount accumulator
Net amount accumulator

COBOL PROGRAM



```
   5                          16.23.56       MAR  3,1978

        008140 PROCEDURE DIVISION.
        008150
        008160*****************************************************************
        008170*                                                               *
        008180*  THIS MODULE CONTROLS THE PROCESSING OF THE INPUT RECORDS,    *
        008190*  THE PRINTING OF THE REPORT, AND FINAL TOTALS. IT IS ENTERED  *
        008200*  FROM AND EXITS TO THE OPERATING SYSTEM.                      *
        008210*                                                               *
        008220*****************************************************************
        009010
        009020 A000-CREATE-CUSTOMER-REPORT.
        009030
        009040     OPEN INPUT  ACCOUNTS-RECEIVABLE-INPUT-FILE
        009050          OUTPUT ACCOUNTS-RECEIVABLE-REPT-FILE.
        009060
        009070*  HEADING ROUTINE
        009080
        009090     WRITE ACCOUNTS-RECEIVABLE-REPT-LINE FROM FIRST-HEADING-LINE
        009100         AFTER ADVANCING TO-TOP-OF-PAGE.
        009110     WRITE ACCOUNTS-RECEIVABLE-REPT-LINE FROM SECOND-HEADING-LINE
        009120         AFTER ADVANCING 2 LINES.
        009130     WRITE ACCOUNTS-RECEIVABLE-REPT-LINE FROM THIRD-HEADING-LINE
        009140         AFTER ADVANCING 1 LINES.
        009150     MOVE SPACE-TWO-LINES TO PROPER-SPACING.
        009160
        009170*  MAIN ROUTINE
        009180
        009190     READ ACCOUNTS-RECEIVABLE-INPUT-FILE
        009200         AT END
        009210             MOVE 'NO ' TO ARE-THERE-MORE-RECORDS.
        010010     IF THERE-IS-A-RECORD
        010020         PERFORM A001-PROCESS-AND-READ
        010030             UNTIL THERE-ARE-NO-MORE-RECORDS
        010040         PERFORM B010-PRINT-FINAL-TOTALS.
        010050     CLOSE ACCOUNTS-RECEIVABLE-INPUT-FILE
        010060           ACCOUNTS-RECEIVABLE-REPT-FILE.
        010070     STOP RUN.
        010080
        010090
        010100
        010110 A001-PROCESS-AND-READ.
        010120
        010130     PERFORM B000-PROCESS-DETAIL-RECORDS.
        010140     READ ACCOUNTS-RECEIVABLE-INPUT-FILE
        010150         AT END
        010160             MOVE 'NO ' TO ARE-THERE-MORE-RECORDS.
```

FIGURE 2

the problem, the next student's design is reviewed. This continues until all three students have had their program design reviewed by the other members of the team.

3. After the program design has been reviewed, each of the students then codes the program. After coding the program, the team then meets again to review the code, examine for errors in syntax, proper implementation of the logic, etc. After all code has been reviewed, only then is the program compiled and executed.

In an unusually high percentage of cases, the instructors will find that students' programs compile and execute properly the first time the programs are run on the computer.

ADVANTAGES OF STRUCTURED WALKTHROUGHS

Using structured walkthroughs in the classroom has a number of decided advantages. Some of these are briefly explained in the following paragraphs.

1. Structured Walkthroughs provide a meaningful learning experience for the students. A great deal of learning can take place in the walkthrough as students have the opportunity to learn from one another; students have the opportunity to teach; and students have the opportunity to view other programmers' approaches to problem solving.

2. Better programs are produced by the students because students know that both their design and code will be reviewed by others; programs are thus made as readable and understandable as possible.

3. Errors will be detected during the walkthrough. Walkthroughs provide the opportunity to correct errors before the testing phase of the program development cycle; and, thus, programs which are submitted for compilation are much more likely to compile and execute properly the first time.

4. Students become exposed to working on a team which tends to improve communication capabilities in addition to improving programming skills.

It should be noted that a structured walkthrough in an academic environment is not limited to a particular language, and is not limited to classes that incorporate structured design or structured programming in their class activities. Structured walkthroughs can, in fact, become an important teaching technique in all programming classes.

THE NEW PHILOSOPHY OF COMPUTER PROGRAMMING

Historically, computer programming as a human endeavor has been approached as being a "natural" error-prone activity in which correct results are produced only after going through an extensive debugging process which involves correction of errors in both the syntax of the language and the logic of the program itself. Far too often in the classroom, instructors have taken the same type of philosophical approach, that is, in teaching computer programming "errors" are considered a natural consequence of a student's activity.

Dr. Harlan Mills, in his paper HOW TO WRITE CORRECT PROGRAMS AND KNOW IT, has stated, "The real objective in programming should be to write correct programs from the start - not merely to emerge from debugging with no errors......Syntax errors are either errors of ignorance or carelessness. If they are errors of ignorance, you need to do more homework on the syntax of your programming language. If they are errors of carelessness, you need to learn how to concentrate and take what you are doing more seriously." (3)

Today, with the new methods of structured design which allow problems to be broken down into a manageable size and structured programming which provides for a more scientific approach to program construction and structured walkthroughs which provide for serious review of one's efforts before compilation and execution of the program, it is now possible to insist upon a new precision in computer programming so that schools will begin producing truly professional programmers to serve the needs of industry in the future.

BIBLIOGRAPHY

1. Dijkstra, E., "GO TO Statement Considered Harmful," Communications of the ACM, Vol. 11, No. 3 (March, 1968).
2. Knuth, Don, "Structured Programming with Go To Statements," ACM Computing Surveys, Vol. 6, No. 4 (1974).
3. Mills, H.D., How to Write Correct Programs and Know It, IBM Federal Systems Division, IBM, (February, 1973).
4. Mills, H.D., "The New Math of Computer Programming," Communications of the ACM Vol. 18, No. 1 (January, 1975).
5. Myers, Glenford J., Reliable Software Through Composite Design, Petrocelli/Charter, (1976).
6. Shelly, G.B. and Cashman, T.J., Introduction to Computer Programming Structured COBOL, Anaheim Publishing Company (1977).
7. Stevens, W.P., Myers, G.J., and Constantine, L.L., "Structured Design," IBM Systems Journal, Vol. 13, No. 2 (1974).
8. Weinberg, Gerald M., The Psychology of Computer Programming, Van Nostrand Reinhold Company, (1971).
9. Yourdon, E., and Constantine, L.L., Structured Design, Yourdon, Inc. (1975).

GENERAL INTEREST

PROCEDURES FOR COLLECTING DATA FROM STATE & LOCAL EDUCATIONAL AGENCIES

Dr. John F. Allan
Dr. John Stiglmeier

Committee on Evaluation & Information Systems (CEIS)
Council of Chief State School Officers (CCSSO)

The Committee on Evaluation and Information Systems (CEIS) is a standing committee of the Chief State School Officers whose purpose is to provide the Council of Chief State School Officers (CCSSO) with a single coordinated effort in meeting present and emerging concerns and in formulating recommendations in the area of federal/state/local educational information systems, including but not limited to research, evaluation, and management information. The paper outlines the origin of CEIS, its organizational structure, and administration procedures. The presentation will be particularly helpful to public and private agencies who desire to collect data from public schools on a state and national level and other organizations who wish to establish procedures for controlling the data burden which is required of all educational systems.

The Council of Chief State School Officers (CCSSO), an organization of superintendents and commissioners of education in the United States and its territories, was established to address the combined interests of its members and their agencies in a wide variety of educational concerns. One specific concern, the recent demands for significant amounts of educational information sought by federal agencies and other national organizations, caused the CCSSO to establish a Committee on Evaluation and Information Systems (CEIS). The purpose of CEIS is to "provide the Council of Chief State School Officers with a single coordinated effort in meeting present and emerging concerns and in formulating recommendations in the area of federal/state/local educational information systems, including but not limited to research, evaluation, and management information."

Organized in 1972, CEIS is made up of representatives of each of the 56 chief state school officers. Activities are directed by an Executive Board consisting of three elected officers and one state CEIS coordinator elected from each of ten regions defined by the Department of Health, Education, and Welfare. Three subcommittees, Data Acquisition, Evaluation, and Information Systems Development, are made up of state CEIS coordinators and other state and local educational agency representatives.

Each subcommittee is responsible for assisting, monitoring, and following the progress of projects related to data collection, evaluation, or information systems that affect state or local educational agencies. In their semiannual reports the subcommittees make recommendations for presentation to CEIS and CCSSO.

DEFINITION OF DATA COLLECTION

Data collection is defined here as any activity that causes information to be maintained by or transmitted from a state or local educational agency to the national level. Data collection activities include but are not limited to requirements to report statistical information, maintain records on individuals or educational programs, compile prespecified data on educational finance, and provide data for research and evaluation studies. Data collection is conducted in various modes: the use of a form developed for a specific purpose; the use of a letter requesting an educational agency to provide certain information; or the administration of a test, questionnaire, or survey form to individual respondents. The planning phase for each data collection activity should include the involvement of persons from state and local educational agencies through CEIS.

FORMS CLEARANCE PROCEDURES

The Federal Reports Act of 1942 as amended requires that specific review procedures be followed by the Office of Management and Budget (OMB) when a federal agency requests information from any nonfederal agency. A federal agency that requests data from any nonfederal agency must follow the guidelines developed by OMB for the management of federal reporting requirements. The sponsoring federal agency is responsible for preparing any documentation for review by OMB.

Section 406 of the General Education Provisions Act (as amended by Public Law 94-482) requires the Secretary of Health, Education, and Welfare and the U.S. Commissioner of Education to coordinate the collection of information and data acquisition activities of the Education Division and the Office for Civil Rights in the Department of Health, Education, and Welfare. The provisions of Section 406 provide an opportunity for state and local agencies to comment on the collection of information and any data collection activity within 30 days of the announced activity. The 30-day notification through the *Federal Register* is intended to inform nonfederal agencies of the data to be collected, the time of the collection, the requirements for a voluntary or mandatory response to the data collection, the respondent's burden in providing the data, and the purpose for collecting the data. Respondents are given a 30-day period to document their concerns about the proposed data collection activity.

Section 406 also requires that any data collection activities requiring information from state or local educational agencies be reviewed by the administrator of the National Center for Education Statistics (NCES). The Educational Data Acquisition Council (EDAC) in NCES serves as the review body for the collection of information and data acquisition activities. Formal clearance procedures established by EDAC and OMB must be followed before any requests are made to collect information from state or local educational agencies. The federal agency, not CEIS, is responsible for official clearance procedures.

CEIS is involved in the review of data collection instruments and in forms clearance to identify redundancies in the collection of information and to determine whether responses constitute a justified burden for state and local educational agencies. The involvement of CEIS should enhance the entire review process and provide OMB and EDAC with an independent review of a proposed collection of information.

PROCEDURES FOR INVOLVEMENT WITH CEIS

An agency sponsoring a data collection activity involving state or local educational agencies should contact CEIS *when the survey design is being planned.* Final action on the survey, including forms clearance, should provide state and local educational agencies with an adequate amount of time before a response is due. In the involvement process the sequence that should be adhered to is as follows:

1. Coordinate data collection activities with the National Center for Education Statistics (NCES) through the Executive Director of the Education Data Acquisition Council (EDAC) to determine (a) if the data are already being collected; or (b) if the need can be merged with similar needs of other agencies for simultaneous collection.

   If the proposed data collection constitutes a unique need, keep NCES informed as the data collection activity is being developed.

2. Contact CEIS through the Joint Data Project staff in the Council of Chief State School Officers (1201 Sixteenth Street, N.W., Washington, D. C., 20036; telephone (202) 833-4207). This contact will result in notification to the appropriate CEIS subcommittee that a new data collection activity is being developed.

3. Communicate with the appropriate CEIS subcommittee.

4. Have the appropriate subcommittee review the design of the proposed data collection activity, using the following criteria:

   a. Statement of purpose and intended use of anticipated outcomes, including any legislative justification
   b. Documentation on unavailability of data requested in existing data collections
   c. Work plan for the activity to determine whether it is practical and manageable
   d. Population to be surveyed, including size of the sample and justification
   e. Presentation of planned layout of data, including table shells, matrices, and analysis plan
   f. Justification and use of each requested item of data.

5. Follow the "Protocol for Data Collection" contained elsewhere in this paper if the activity has been recommended by CEIS.

CEIS subcommittees meet periodically to review data collection activities with persons from the sponsoring agencies. Although the review and analysis of the proposed information collection does not constitute CEIS endorsement of the activity, the review should lead to one of the following positions:

Recommend: Both the concept and instrumentation of a study or survey have been reviewed and recommended; no further action is required by the subcommittee.

Pending: A data acquisition concept has been reviewed; however, no recommendation is given until final drafts of the survey instruments have been reviewed.

No position: A position of neutrality is decided on; the subcommittee will not work for or against implementation of a project.

Not recommended: Either the concept or the instrumentation is not acceptable to the subcommittee. The data collection activity should not proceed in state or local educational agencies.

Although a record of CEIS positions is transmitted to all state CEIS coordinators, EDAC, and OMB, the CEIS position on data collection activities is not binding on any state or local educational agency. Each state or local district has the prerogative to determine its own participation in a data collection activity. Each chief state school officer may propose to supply the requested information directly to the sponsoring agency, may wish to coordinate the study within the state, or may decide to have the sponsoring agency go directly to the selected local educational agencies involved.

PROTOCOL FOR DATA COLLECTION

To ensure adequate cooperation and communication between agencies or their contractors and the data suppliers in state or local educational agencies, CEIS should be involved in the planning stages of the study. After CEIS has completed its review of a data collection activity and other appropriate federal clearances have been obtained, the following protocol should be observed:

1. The chief state school officer (CSSO) should be informed of any data collection activity to be conducted in his or her state *before* the study or collection is made. The CSSO should be informed of the voluntary or mandatory nature of the activity and, if applicable, should be offered the opportunity to manage the collection of data. Any chief state school officer may elect to participate in the study or may choose not to become involved. A copy of the letter sent to the CSSO should also be sent to the state CEIS Coordinator. An up-to-date list of state CEIS coordinators is available from the office of the Council of Chief State School Officers.
2. Any letter to the chief state school officer about a data collection activity should indicate whether or not the study is being coordinated with CEIS and is being reviewed by CEIS representatives. The letter shall not indicate CEIS support of the study until CEIS makes its recommendations in writing to the agency.
3. Direct contacts with a local educational agency should be made by the chief state school officer unless otherwise directed by the chief state school officer. Local educational agencies should be informed of the voluntary or mandatory nature of the activity. The state agency may be able to provide the data from established files or may wish to collect the data.
4. All correspondence with the chief state school officer about a study should be signed by an agency official, never by a contractor. Copies should always be sent to the state CEIS coordinator.

The CEIS chairperson, the CEIS Executive Board, and the Joint Data Project staff coordinate the assignment of data collection activities to one of the three subcommittees.

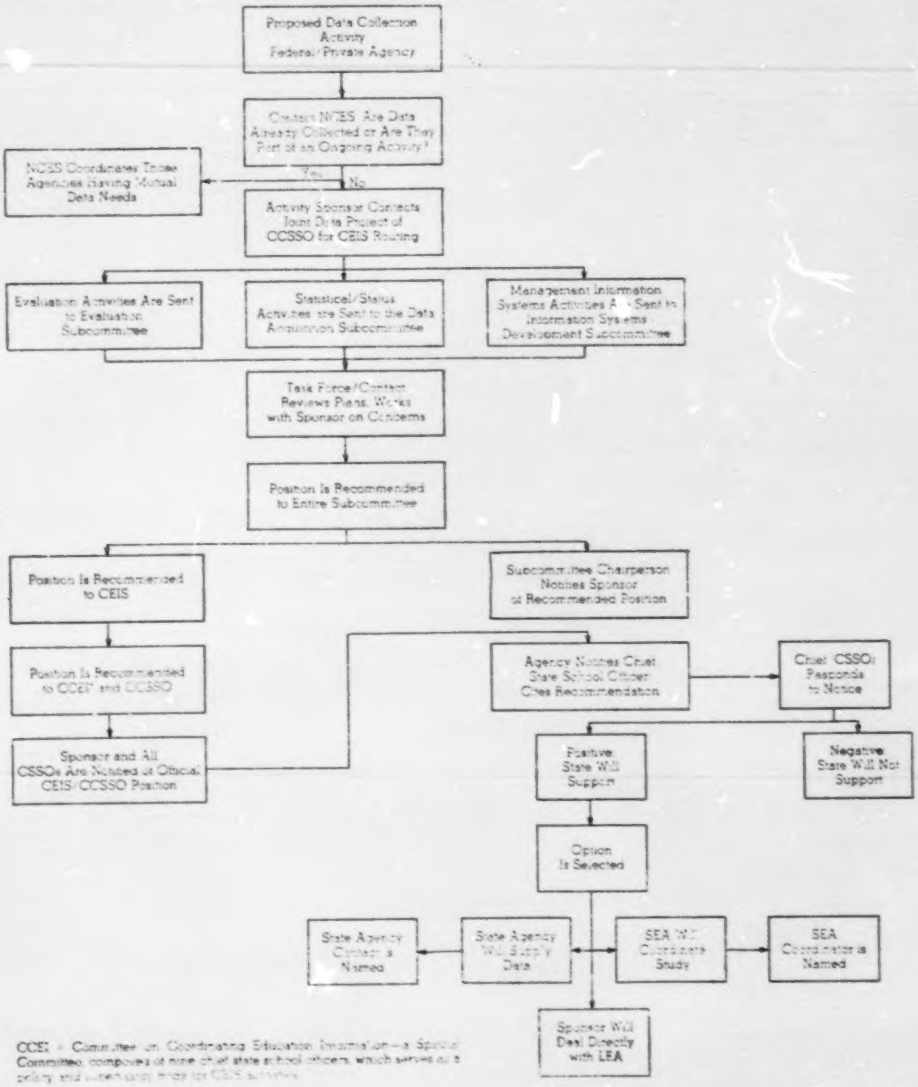
The Data Acquisition Subcommittee is responsible for reviewing proposed activities that involve collecting basic statistical data in response to one-time or recurring requests.

The Evaluation Subcommittee is responsible for monitoring agency studies that (1) are identified as evaluation studies by the agency; (2) are conducted over a designated period of time to provide specific information; (3) do not require continual reporting of data by the respondent agencies; (4) are oriented to research or evaluation as contrasted to the provision of statistical information; or (5) are intended to examine the outcomes or effectiveness of a specific program.

The Information Systems Development Subcommittee plans and initiates appropriate activities to aid in the development and implementation of information systems that serve state and local needs as well as federal reporting requirements. The subcommittee also cooperates in the development of long-range statistical collection systems. Generally speaking, the subcommittee does not monitor current operational data collection activities.

The Council of Chief State School Officers, through CEIS, seeks to participate continually in the planning for collection of educational data by federal agencies or national organizations. This participation requires the involvement of CEIS from the early stages of planning until a recommendation by CEIS is reached. Although the extensive participation of CEIS in the planning stages of a data collection activity can be time-consuming, the CEIS effort will result in improved reliability of the data collected and a reduced work load for state and local educational agencies.

Procedure for Interaction with the Committee on Evaluation and Information Systems (CEIS)


Proposed Data Collection Activity Federal/Private Agency
Contact NCES: Are Data Already Collected or Are They Part of an Ongoing Activity?
Yes
No
NCES Coordinates Those Agencies Having Mutual Data Needs
Activity Sponsor Contacts Joint Data Project of CCSSO for CEIS Routing
Evaluation Activities Are Sent to Evaluation Subcommittee
Statistical/Status Activities are Sent to the Data Acquisition Subcommittee
Management Information Systems Activities Are Sent to Information Systems Development Subcommittee
Task Force/Contact Reviews Plans, Works with Sponsor on Concerns
Position Is Recommended to Entire Subcommittee
Position Is Recommended to CEIS
Subcommittee Chairperson Notifies Sponsor of Recommended Position
Position Is Recommended to CCEI* and CCSSO
Sponsor and All CSSOs Are Notified of Official CEIS/CCSSO Position
Agency Notifies Chief State School Officer Cites Recommendation
Chief CSSO Responds to Notice
Positive State Will Support
Negative State Will Not Support
Option Is Selected
State Agency Contact is Named
State Agency Will Supply Data
SEA Will Coordinate Study
SEA Coordinator is Named
Sponsor Will Deal Directly with LEA

*CCEI = Committee on Coordinating Education Information—a Special Committee, composed of nine chief state school officers, which serves as a policy and supervisory body for CEIS activities.

COMPUTER ASSISTED RESEARCH: A BIBLIOGRAPHIC DATA BASE FOR LOCATING EARLY MUSICAL SCORES BY BLACK AMERICANS

John M. Eddins and Samuel A. Floyd, Jr.
Black Music Research
Southern Illinois University
at Carbondale

A computerized index of musical scores by black American composers before 1910 is described. A set of scores collected for possible publication in an anthology of music by black American composers served as the initial data base. Fully implemented index files will contain several thousand records, with separate items for composer, title, publisher, place, date, repository, performing medium, and comments. Records can be searched by any item, keyword, or combination, and reports can be generated on any standard output device, in any format. Searching and report generation are user controlled, through interactive computer terminals. The index will provide interested scholars with an efficient means for discovering what musical scores exist and where they may be found. An accumulated data base sufficient for limited use should be available by fall, 1978.

A computerized index of musical scores by black American composers is being developed at Southern Illinois University at Carbondale, as part of the program in black music research. The first stage of this effort is supported by the University through a special research award. The need for the index was made evident through work on another project; development for publication of a four-volume anthology of musical scores and sheet music by black American composers (Floyd, Note 1). From the beginning of the anthology project, two main problems were encountered; (1) securing musical scores and sheet music by composers whose racial identity, as black, had already been established, and (2) determining which of the many hundreds of nineteenth and twentieth century American composers were of black heritage. The problems were especially acute for the nineteenth century composers. It soon became evident that these problems will continue to present major difficulties for researchers in the field, even after current work on the anthology (and related projects by other scholars) is completed. To help future researchers, it was decided to use the information already obtained from the anthology project as the basis for a data bank containing bibliographic information, location, and descriptive comment on all known musical scores written by black Americans before 1910. The purpose is to provide interested scholars with an efficient computer indexing system for discovering what musical scores exist and where they may be found.

One might think that all published materials needed to develop an anthology such as the one currently being prepared would be available and easily accessible at the Library of Congress. This, unfortunately, is not the case, for three main reasons; (1) many of the compositions were not registered at all in spite of the fact that they carry copyright notices, (2) many are on hand at the Library of Congress, but in reserve storage, due to a lack of shelf space and to the fact that they have not been catalogued, and (3) many of the pre-1900 compositions which were registered with circuit courts never were delivered, as requested, to the Library of Congress. The latter situation is strikingly illustrated by those compositions published in New Orleans prior to 1870, most of which never were delivered. Problems in locating and obtaining these compositions make it difficult for scholars interested in nineteenth century black composers in New Orleans to carry out effective research; yet New Orleans was a place where there was significant artistic freedom for free blacks.

Even if all the compositions were available at the Library of Congress, they would remain inaccessible to many scholars because of distance and travel limitations; however, a number of repositories across the country have significant holdings of scores and sheet music by black American composers. For example, several repositories other than the Library of Congress hold most of the publications of black composers who were active in Philadelphia during the early nineteenth century; Frank Johnson, James Hemmenway, Isaac Hazzard, A. J. R. Conner, and William Brady. Several repositories also hold scores by James P. Johnson, Eubie Blake, and other early twentieth century composers. There is presently no way, however, for a researcher to determine all of the various locations of these scores without a great deal of preliminary research.

The computerized index now being developed will eliminate the need for such preliminary research, thus saving scholars valuable time and effort. Researchers will be able, through the service provided by this tool, to locate materials in repositories near where they live, making much expensive travel unnecessary. The need for this tool is especially acute since the new copyright law severely restricts photo-duplication of published materials.

The Remote Information Query System (RIQS), developed at Northwestern University (Borman, Chalice, Dillman, Dominick, and Kobbe, 1974), was chosen for computer software. RIQS is a data-base system capable of managing both text and numeric information. It was designed so that a user with little training as a programmer can structure records to suit special needs, create files, and retrieve and analyze data through convenient input-output modes. Three considerations in particular led to the choice of the RIQS system for this project:

1. Record structures, search procedures, and report generation are controlled by the user, and can be tailored to individual needs.

2. Processing is done interactively, at a terminal, providing speed, convenience, and flexibility.

3. The system is powerful, yet simple to use. No intermediary search analyst is needed; only some elementary programming skill, plus the ability to use a typing terminal.

A file structure has been developed for the bibliographic index, in which each record contains separate items for composer, title, publisher, place, date, performing medium, type of music, locations of copies, and comments. The date item is a number; the other items are free-field character strings. For search purposes, any word or group of words in a character string can be treated as a keyword. Records can be searched by any combination of items and keywords, and reports can be generated on any standard output device, in any format desired.

Searches can include all records in the file, or only a set of specific records. Search commands as well as report contents and format are specified by the user, in relatively simple, user oriented language.

To interrogate the data base, one ordinarily signs on to a terminal, calls the file, specifies the search conditions and the form of output, and immediately receives the results at the terminal. Output also can be diverted to a line printer, or to tape or disc storage. Search commands may include specification of the record or set of records and the item or items to be searched, plus logical operators to define the search conditions. For instance, to request all scores by Dabney, the user would type:

```
IF #1 EQ 'DABNEY'
```

The search condition will be satisfied if the composer's name (item #1) equals (EQ) or contains the word "DABNEY." Multiple conditions can be specified:

```
IF (#1 EQ 'HEMMENWAY' AND #3 EQ
'PHILADELPHIA') AND #6 EQ 'PIANO' AND
#5 (GE 1820 AND LT 1830)
```

Here, the conditions are satisfied only if the composer is Hemmenway, the score is for piano, and was published in Philadelphia between 1820 and 1830.

The following code requests the titles and repositories of all scores in the file by James P. Johnson (the word "ZAP" causes suppression of item numbers, names, and descriptions in the report):

```
IF #1 EQ 'JAMES P. JOHNSON'
DISPLAY ZAP #2 #8 END
```

In addition to the items to be listed, output report specifications may include format control and insertion of text, such as headings.

```
DISPLAY ACROSS ZAP #1 #2 #3 #4 #5
SKIP 1 'KIND OF MUSIC --' #7 SKIP 1
'PERFORMING MEDIUM --' #6 SKIP 1
'SCORE LOCATIONS --' #8 SKIP 1
TAB 15 'COMMENTS' SKIP 1 #9 END
```

This set of commands would cause those records satisfying some specified search conditions to be printed as follows:

```
HEMMENWAY, JAMES.  "THE FIFTH SET OF
QUADRILLES".  PHILADELPHIA, GEORGE
WILLIG, 1828.

KIND OF MUSIC -- DANCES

PERFORMING MEDIUM -- PIANO

SCORE LOCATIONS -- LIBRARY OF CONGRESS
```

COMMENTS

OF THE FIVE SETS OF QUADRILLES COMPOSED BY HEMMENWAY, ONLY THE LAST FOUR HAVE BEEN LOCATED. THE SECOND SET WAS PUBLISHED BY WILLIG IN 1825. THE THIRD AND FOURTH SETS WERE PUBLISHED BY WILLIG IN 1827. SEE LIBRARY OF CONGRESS COPYRIGHT RECORDS. SCORES ARE HELD BY THE LIBRARY OF CONGRESS, MUSIC DIVISION, M1.A13H.

Other features of the RIQS system include routines to do counts, sums, means, averages, and other computations using variables, loops and arithmetic operators. Any items in a record can be declared multiple, and can be searched by sub-items, such as #8.1, #8.2, #8.3, etc. Any search routine can be defined as a procedure, catalogued, and called again as needed, so that the user does not have to reconstruct it repeatedly. Records can be assigned to one or more logical sets, and searches can be limited to selected sets -- a convenient means for reducing search time when the file becomes very large. Any item may be specified as an indexing field for sorted output.

Sufficient funds currently are being sought for the extensive repository searches which the project will entail. with a target date for completion in about two and a half years. When fully implemented, the index files will contain several thousand records. Stored information in the index will be made available to scholars by mail requests and through local on-line access. It is anticipated that an accumulated data base sufficient for limited use will be available by late fall, 1978.

Reference Notes

1. Floyd, S. A. The music of black American composers: An anthology (National Endowment for the Humanities, Project No. RT-24919-76-808). To be published in four volumes by the Southern Illinois University Press.

References

Borman, L., Chalice, R., Dillman, D., Dominick, W., and Kobbe, R. RIQS Remote Information Query System Users Manual. Evanston, Ill.: Northwestern University, 1974.

BRAILLE AND THE COMPUTER

JOHN GEIL
REGION IV
EDUCATION SERVICE CENTER

The need for more innovative methods of providing braille to the blind student is explored and a need for more current and a wider variety of brailled material established. The history of computer generated braille at the Region IV Education Service Center, Houston, Texas is reviewed. An innovative computer assisted instruction program for the School for the Blind in Austin, Texas involving inter-active braille embossing terminals linked to the Region IV Computers in Houston is discussed. The CAI program involves conversion of the Suppes CAI programs to grade II braille along with the Region's secondary CAI programs. Incorporated in the project are a number of innovative approaches to vocational training.

The need for more current and a wider variety of Braille material for the blind becomes more apparent each year as new methods and techniques open up new and more varied employment opportunities for the blind.

Paul Fortier and Donald Keeping of the University of Manitoba in a research paper, "Another Face of Computing; Service to the Blind," state ----the amount of available Braille has never approached fulfilling the need. In the past few years a number of circumstances have converged to bring this paucity of Braille to crisis proportions. First there is a growing problem of personnel. Traditionally, producers of Braille have been either volunteers working without remuneration, or people willing to work for low wages which were all that charitable organizations could afford to pay. Recent social changes have resulted in the virtual disappearance of volunteer labour, sharp increases in the minimum wage, and a reduction of the number of people willing to carry out a difficult and complex task for low wages.

Increases in the price of raw materials have meant that larger and larger press runs are required before production of multiple copies from zinc plates becomes cost effective. Storage of the zinc plates represents both an immobilization of capital and a serious warehousing problem.

Finally, it has been realized that the facilities of special residential schools for the blind are better reserved for blind students having multiple handicaps. Blind students without other afflictions are better off in ordinary schools where they can learn to function in a sighted society, and where they are not cut off from the goals and aspirations to which their abilities entitle them. This, however, requires a wide range of Braille texts, because textbooks vary from one school division to another, and textbooks are constantly being changed or revised. This was not a problem when all blind students were in a special school, where it did not matter if the texts were out of date, so long as they were the same for all the students. Given the variety of texts needed and the short time usually available between the finalization of a textbook list and the beginning of classes, traditional methods cannot produce sufficient Braille, in a short enough time, to meet the need.

Thus, traditional Braille production methods are not able to meet the basic needs of primary and secondary education of the blind. Braille material for other purposes--like technical manuals or university texts--is even more difficult to obtain." Not mentioned in the report are the unavailability of other items which are almost a necessity to the sighted such as bank statements, employment openings, etc.

In 1975 Region IV Education Service Center undertook a review of the textbook system in use in various districts in Texas with the objective of offering suggestions for increasing their effectiveness. Throughout the review recurrent problems with Braille texts were indicated and generally corroborated the findings of Fortier and Keeping. Among

these were:
(1) Delays in receiving Braille material.
(2) Receiving the back sections of a Braille text three months before receiving the front sections.
(3) Unavailability of certain texts in Braille.
(4) Lack of current Braille material.
(5) Bulk of Braille volumes and storage space required to house them.

These problems led to a study of the reading requirements of the blind or visually handicapped student. The study revealed the following conditions:
(1) The blind student has available only a limited variety of reading material.
(2) Extensive research or outside reading requires the employment of a human "reader," due to the limited availability of Brailled material.
(3) The elapsed time between the discovery of the requirement for a Brailled publication and its availability can be 12 months or longer.
(4) The blind student is further handicapped in that reading material of a current nature is not generally available to him in Braille form.
(5) Under present methods, the variety and timeliness of Braille material is likely to be further reduced as fewer Braillists are being trained each year.

The study then shifted to methods employed in the preparation of Braille material. The following determinations were made:
(1) The manual process of transcribing Braille texts is slow and expensive. Elapsed time for transcription of a text can be as much as a year.
(2) Computer assisted Braille texts are prepared by:
  a. Atlanta School District, Atlanta, Georgia.
  b. Printing House for the Blind, Louisville, Kentucky.
  c. A few other non-commercial organizations.
(3) Computer programs and technology exist to:
  a. Keypunch a printed text, introduce the data to computer tape, and convert the data to Braille language, thus:
    i. Print a text by computer, or
    ii. Print a text by an automatic Braille printer.
  b. Type a text and create a cassette tape, introduce the tape to a computer, and convert to Braille and:
    i. Create a translated cassette tape for later embossing in Braille, or
    ii. Drive an embosser to Braille a text.
  c. Obtain a publisher compositor tape, introduce the tape to a computer, convert it to Braille language and
    i. Drive a page embosser creating a text, or
    ii. Create a translated Braille tape to drive a Braille press.

The need for more current and a wider variety of reading material for the blind and visually handicapped was certainly evident. Moreover, it appeared that the blind student could benefit from participating in computer assisted instruction programs. It further appeared that a wider variety of vocational opportunities were needed by the visually handicapped. In addition, computer generated Braille appeared not only feasible but cost effective.

Therefore, in the fall of 1976 the management of Region IV decided to make a substantial investment in hardware and talent to pioneer in the development of computer assisted techniques to aid the handicapped. Initial efforts were to be focused on the visually handicapped with some cursory investigations into the problems of the hearing handicapped. The Region proceeded to acquire a Braille translation computer program, and interactive tape embosser, and a number of other related hardware items.

The Regions plans and objectives (detailed by the charts in Exhibit A can be summarized as follows:
(1) Develop computer assisted braille II translation capabilities to:
  a.Accept and translate typewritten input
  b. Accept and translate foreign languages
  c. Accept and translate publishers compositors tapes
  d. Accept and translate computer assisted instruction programs
  e. Accept and translate news wire releases.
(2) Test and evaluate various interactive embossing devices.
(3) Establish a cassette library of translated braille reading material.
(4) Provide braille textbooks and other student reading material from typewritten input and publishers compositors tapes.
(5) Develop additional vocational opportunities for the visually handicapped.
(6) Establish a daily news release capability on accessible computer disc.
(7) Install a computer assisted instruction capability in braille.
(8) Develop an interactive braille capability for the Texas School for the Blind and other schools for the visually handicapped.

Early in 1977 the Region received some assistance for the program in the form of a small grant from the Governor's Coordinating Office for the Visually Handicapped. With these funds and some additional support from Region IV the project was able to demonstrate a satisfactory braille II computer translated braille capability. From January through May the following projects were initiated:

A. School Textbooks - The elementary textbook "It Happened This Way" has been placed on cassette tape and translated. It is ready for publication once certain hardware modifications have been made to the braille embosser. (The modifications are now complete.) The embosser generates what the manucalls standard braille which deviates slightly from Library of Congress braille.

B. Outside Reading Material - The minutes of the "AD HOC Advisory Committee of the Visually Handicapped" have been placed on cassette tape.

C. Computer Assisted Instruction - CAI programs are currently operable in grade I braille. They are expected to be operable in grade II braille by September if funding is available.

D. News Wire Service - Arrangements are being made with the Houston Post Newspaper for a direct wire tie in to the Region IV computer. The computer file will be updated daily representing a mini-newspaper and processed through the translation program. Among other things the mini paper will contain World, National, State and City news summaries; Sport Scores; Weather; Reviews; Music/Theatre and Book Reviews; People; Commentary; Lynn Ashby; Names in the News; Today's Prayer; and Astrocast. The file will be accessible from any remote location by a braille embossing terminal.

E. Vocational Applications - The Texas Bankers Association series entitled "Your Bank and Personal Money Management" is being reviewed for conversion to a computer assisted instruction (CAI) program.

During the latter part of May 1977, a small supplemental grant was received from Texas Education Agency to support the braille project through the month of June. During this period a more detailed review of problems associated with the translation of compositors tapes was conducted.

It was determined that compositors tapes are generally magnetic tapes or punched paper tape using a variety of bit structures and varied languages. This lack of standardization represents one of the principle problems to be resolved in translation. Another is the lack of change control and identification resulting in composition changes not reflected on the compositor tape. A secondary problem will be the "culling" of extransous press instructions from the text material. Arrangements are being made with Control Data Corporation to assist in resolving these problems.

A cursory review was also made of the computer assisted instruction requirements of the Texas School for the Blind. As a result a proposal was prepared titled "Statewide Computer Braille Proposal" proposing the installation of interactive braille embossing terminals at the School as a pilot project. The terminals were to be "on line" with the Region IV computer in Houston making computer assisted instruction available to the students at the at the School. The Suppes CAI packages would be offered to the school along with the braille translation program, the mini newspaper, the computer games, the Region IV braille "library" and the occupational data bank.

In anticipation of acceptance of the proposal to the School for the Blind, discussions with Computer Curriculum Corporation were initiated on modifying the translation program to accomodate CAI in grade II braille.

Additional sources of revenue are being sought to expand these and other projects.

On November 1, 1977 Region IV was awarded a substantial grant to implement their proposal for the Texas School for the Blind and continue their Research and Development program.

The proposal consisted of four principle projects:(1) Implementation of a pilot Computer Assisted Instruction program at the School for the Blind in Austin (2) provision of computer generated braille textbooks and supplemental reading material; (3) continuing research and development activities as part of the project and (4) evaluation of the pilot Computer Assisted Instruction program at the School for the Blind and expansion of the program as a statewide service.

The proposal will make available to the Texas School for the Blind a pilot installation of interactive computer terminals designed for the visually handicapped student and a variety of computer assisted instruction programs including the widely recognized Suppes Programs. In addition the full resources of the most powerful educational computer facility in the United States (Exhibit B) will be available to the school and its students. Among these resources are the following:

- Translated library of braille reading material.
- Braille translation program (Grade II Braille).
- Daily News File (Houston Post Newspaper).
- Student Progress Reports.
- Computer Assisted Instruction Programs
- Vocational Applications
    - Self Teaching Courses in Computer Programming
    - Occupational Data Bank.
- Management Programs (Available at extra cost.)
    - Student Scheduling

Grade Reporting
Average Daily Attendance
Games (Star Trek, Chess, etc.)

The proposal was based on a recently completed review of the needs of the Texas School for the Blind by our Computer Assisted Instruction consultant. It was determined that 135 children could benefit from Computer Assisted Instruction Programs and related braille material. On this basis six (6) plus terminals would be required to accomodate the Suppes Computer Assisted Instruction Programs, leaving a terminal partially free for textbook, news release and other activities.

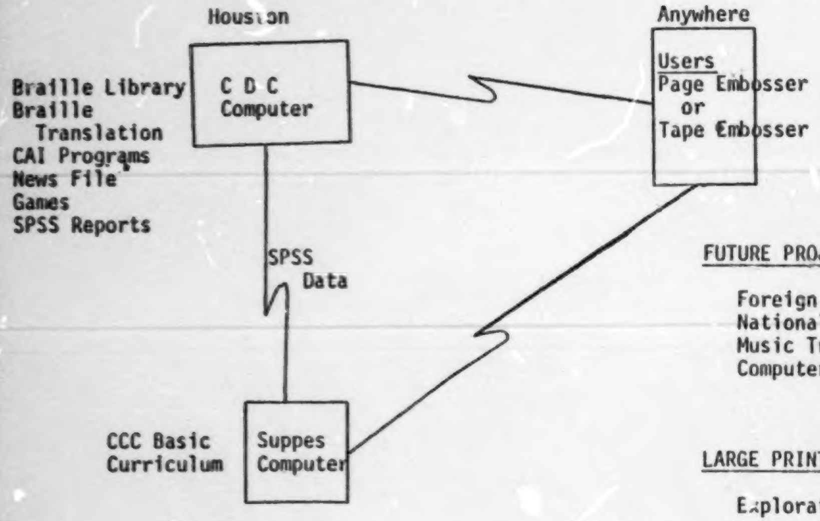
In coordination with the Textbook Division of the Texas Education Agency the Region further proposed to provide Computer Generated Braille textbooks and designated supplemental reading material. Concurrently with the first two projects the Region further proposed to maintain an innovative and active research and development program to advance the state of the arts in conjunction with a realistic evaluation program.

Due to the interrelationship of elements within the various projects, the program contemplated concurrent development of facets of the various projects. Therefore, both planning and costs are interrelated and not clearly segregated between the projects. Work is currently proceeding in all of the project areas.

REGION IV EDUCATION
SERVICE CENTER

EXHIBIT A

COMPUTER ASSISTED INSTRUCTION


Houston
Braille Library
Braille Translation
CAI Programs
News File
Games
SPSS Reports
C D C Computer
Anywhere
Users
Page Embosser
or
Tape Embosser
SPSS Data
CCC Basic Curriculum
Suppes Computer

VOCATIONAL APPLICATIONS

- Computer Programming
- Banking
- Brokerage
- News Casting
- Publishing
- Employment Agencies
- Weather Bureau
- Government Agencies
- Ministry

FUTURE PROJECTS

- Foreign Language Translation
- National Network of Braille Cassette Libraries
- Music Translation
- Computer Generated Embossed Graphics

LARGE PRINT

Exploratory

- Computerized Variable Type Size Program
- Laser Beam Recorder to Microfiche to Large Print

DEAF AND HARD OF HEARING

Exploratory

Computer Assisted Instruction

Total Communication Employing CRT's and Computer Driven Microfiche.
CAI Programs Oriented to the Deaf.

OBJECTIVES

BLIND AND PARTIALLY SIGHTED

BRAILLE

Texts and Reading Material

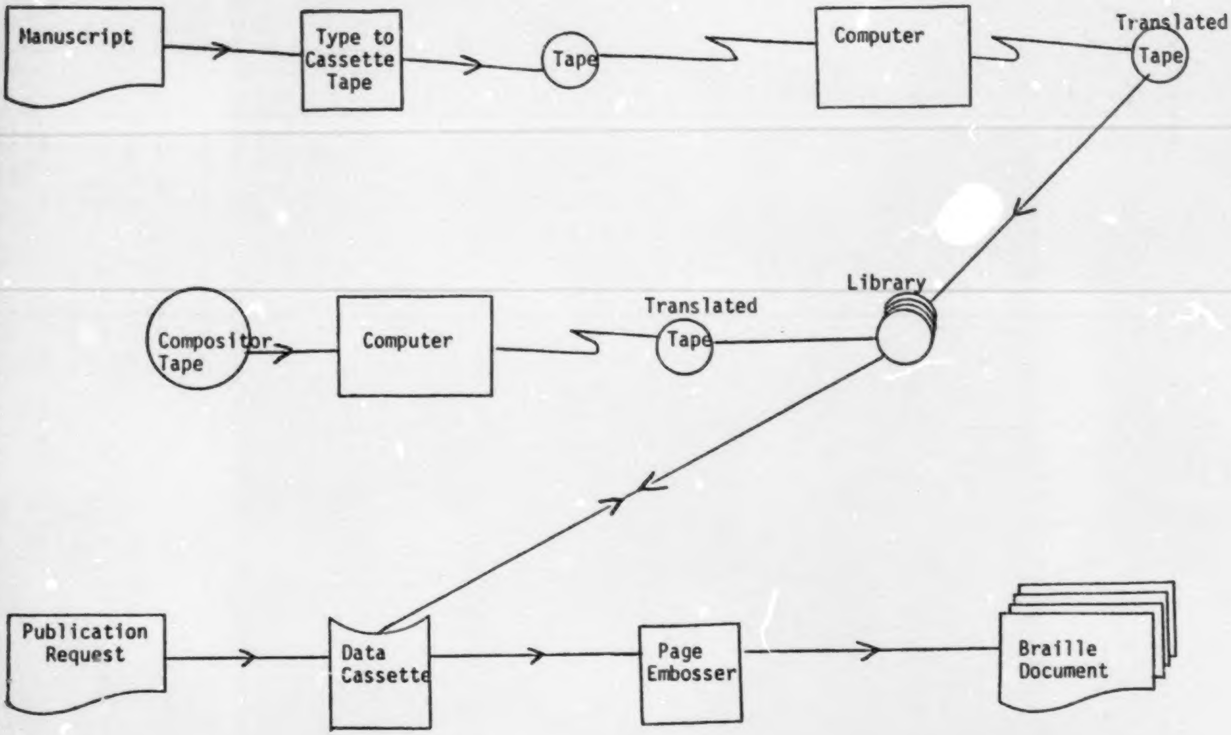
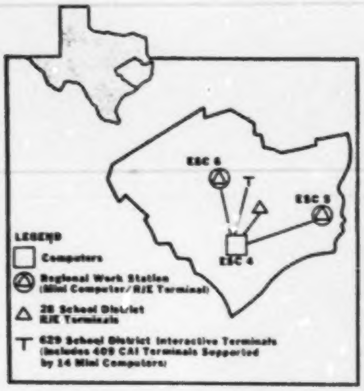

Manuscript
Type to Cassette Tape
Tape
Computer
Translated Tape
Library
Compositor Tape
Computer
Translated Tape
Publication Request
Data Cassette
Page Embosser
Braille Document

EXHIBIT B

GULF TEXAS MULTI-REGIONAL PROCESSING CENTER

REGIONS 4, 5, AND 6

The Gulf Texas MRPC serves school districts in three education service center regions. This cooperative computer processing facility is operated by the Region 4 Education Service Center in Houston. Administration of the MRPC is performed through a Management Committee dedicated to providing quality, low cost data processing services.


ESC 6
ESC 5
ESC 4
LEGEND
Computers
Regional Work Station (Mini Computer/RJE Terminal)
28 School District RJE Terminals
629 School District Interactive Terminals (Includes 409 CAI Terminals Supported by 14 Mini Computers)

MANAGEMENT COMMITTEE
Wesley Summers, Bryan ISD (Chairman)
D. E. Bailey, Region 5 ESC
Gerald Cobb, Galena Park ISD
Clyde Gott, Port Arthur ISD
T. S. Hancock, Region 4 ESC
Elliott Lansford, Splendora ISD
Glen Pearson, Bridge City ISD
Max Schlotter, Region 6 ESC
Linus Wright, Houston ISD

OPERATIONS COMMITTEE
Arley Carter, Region 4 ESC (Chairman)
(713) 621-9490
Jim Brod, Houston ISD
Rex Williams, Spring Branch ISD
R. A. Hidalgo, Region 5 ESC (713) 892-9562
Frank Lown, Bryan ISD
Charles Park, Region 6 ESC (713) 295-9161
Rex Schellenberger, Region 5 ESC (713) 892-9562

DATA PROCESSING DIRECTOR
Arley Carter
P. O. Box 863
Houston, Texas 77001

	REGION 4 HOUSTON		REGION 5 BEAUMONT	REGION 6 HUNTSVILLE	TOTALS GULF TEXAS MRPC
PARTICIPATION					
Student Files	323,292		49,332	25,957	398,381
Scheduling (Students)	202,387		29,835	8,295	240,517
Grade Reporting (Students)	210,776		29,118	8,295	248,189
Attendance (Students)	423,798		33,993	25,694	483,485
Payroll Files (Personnel)	55,387		5,659	5,271	66,317
Payroll Accounting (Checks)	903,155		57,180	56,680	1,017,015
Test Scoring (Tests)	313,055		35,926	36,992	385,973
Financial Accounting (Districts)	18		4	13	35
Tax Accounting (Districts)	13		3	4	20
Personnel Accounting (Districts)	6		1	1	8
Interactive Terminals:					
Problem Solving	129		2	-0-	131
Computer Assisted Instruction	349		30	30	409
Administrative	89		-0-	-0-	89
Vocational & Administrative RJE Terminals	25		2	1	28
Total Districts Served	42		22	42	106
RESOURCES	Region 4 ESC	Houston ISD			
NUMBER OF PERSONNEL	79	38	7½	4	128½
HARDWARE:	Computers	Computer	Terminal	Terminal	
Make/Model	Two CDC CYBER 172-8	IBM 370/145	Cincinnati Milacron CIP/2000	Cincinnati Milacron CIP/2000	
Memory	262K Words (512K Words Extended Core)	256K Bytes	32K Bytes	32K Bytes	
Printers	Two 1200 LPM	Two 1100 LPM	600 LPM	600 LPM	
Card Reader	One	One	Mark Sense	Mark Sense	
Disk Drives	Eighteen 844-2	Eight 2314	One		
Tape Drives	Eight	Four	One		
Scanner	NCS Sentry 70				
Mini Computers:					
Control Data Corporation	CDC 734	CDC System 17 Two 1200 LPM Printers, Two Tape			
Cincinnati Milacron	CIP/2000 32K Byte Memory, Mark Sense Readers, Tape, and Disk	CIP/2000			
1976-77 Budget					$4,431,600

A STATEWIDE COMPUTING CONSORTIUM: ITS PROGRESS AND PLANS

John E. Haugo
Minnesota Educational
Computing Consortium

The Minnesota Educational Computing Consortium (MECC) is an organization created by the four public educational systems in Minnesota to coordinate and provide computer services to students, teachers and educational administrators throughout the state. MECC draws upon the resources of member systems and 75 professional staff in providing the overall review of computing plans and budgets, a statewide instructional computing system, the development of regionally-based management information systems and support to a variety of special projects utilizing computers.

The Minnesota Educational Computing Consortium (MECC) is an organization that transcends the various organizational levels of education, serving elementary-secondary, vocational-technical, and higher educational institutions. MECC is able to do this because the State of Minnesota has a joint powers law. Under this law, governmental units can agree to form an organization which cooperatively exercises the powers common to each of the entities forming the organization. In the case of MECC, the entities which entered into the agreement are: the Minnesota State University System, the Minnesota Community College System, the University of Minnesota, the State Department of Education, and the State Department of Administration. The latter agency serves as a member because of its statutory authority relative to State University and Community College systems.

MECC has the dual goals of coordination and service. In its coordinator role, MECC maintains a long range master plan and a biennial plan for educational computing in the state, and reviews and assists institutions in developing their computer plans in light of these plans. In its service role, MECC operates computing facilities to provide instructional computing, acts as a broker for similar services from member institutions, develops administrative software and assists in its installation, and contracts for computer equipment that can be used by its members.

HISTORY

MECC was formed in 1973. Prior to its formation however, there had been a history of cooperative computing activities and studies of computing needs for the state.

TIES, a regional data processing cooperative for thirty metropolitan school districts, was already operational. The Minnesota Educational Regional Interactive Time-Sharing Systems (MERITSS) under the operating management of the University of Minnesota was providing interactive instructional services to the university, state community colleges, two state universities, some private colleges, and several school districts. The Southern Minnesota School Computer Project (SMSCP) was also operational with Mankato State University supplying instructional computing services to several school districts. In addition, the State University System had a plan for an academic/administrative network. Committees to review computer budget requests had been formed in 1969 and 1971 by the Minnesota Higher Education Coordinating Commission. Other extensive studies with considerable user advice had determined the computing needs at all educational and state and local governmental levels.

Early in 1971, the Governor issued an executive order which declared a moratorium on further computing expenditures. He appointed a nine member joint committee of chief executive and administrative officers representing the Minnesota public and private educational systems, the Governor's office and the Department of Administration. The committee's charge was to review the outputs of the previous studies and determine the possibility of a statewide cooperative educational computing effort.

The committee considered five alternative organizational structures to accomplish the cooperation desired and selected the joing powers approach. These alternatives had been prepared by a six member task force appointed by the committee. The task force was expanded in size to twelve and instructed to carry the plan back to the educational sectors to seek their input. This effort resulted in the development of a report entitled, "A Proposed Educational Computing Services Organization: Its Facilities and Services," which presented the proposed organizational structure and goals for MECC together with a proposed biennial budget. The plan received legislative support in 1973 and MECC was formed.

ORGANIZATIONAL STRUCTURE

MECC has approximately seventy-five staff members and is governed by a sixteen-member Board of Directors. The State Board of Education appoints six members to the Board. In the past these members have been representatives of the State Department of Education, and local school district administrators and board members. The three systems of higher education each appoint two Board members. These members have included high level administrators from the system offices and governing board members. The Commissioner of Administration appoints one member and the Governor appoints three members to the Board. Traditionally, one of the Governor's appointees has been from the private college sector. The Board of Directors establishes overall policy and reviews the operation of the consortium.

The consortium also has an advisory structure consisting of a Planning and Budgeting Committee and a Facilities and Services Review Committee. The latter group is responsible for review of all proposals for computer services. Additionally, there are a number of standing user advisory committees and periodic ad hoc task forces.

One of the major premises upon which MECC was formed was that it be user controlled. Consequently, the MECC staff actively seeks input from the advisory groups when establishing policies and procedures. Likewise, MECC provides services and the staff feels that they must have the direct input of the users of their services. For example, the educational secondary or vocational school districts around the state are divided into seven regions. Each region has its instructional advisory group which interacts directly with the MECC staff on the interactive instructional time sharing services it provides.

FUNDING

Funding for MECC has come primarily from state sources. Almost all of the budget is either appropriated to the four educational systems or comes from the school districts. MECC signs an annual agreement with the educational systems to provide specific services. The level of services and amount of the contract are fixed prior to the system's funding request to the legislature. This type of agreement provides around eighty percent of MECC's total budget requirements. MECC also provides instructional timesharing services to the school districts. These are billed to the school districts at a fixed annual user charge per terminal access. This source of funding represents slightly less than twenty percent of MECC's total budget. Other support for special projects comes from sources external to the State.

GOALS

The primary goal of MECC is to assist its member systems in the cost/effective application of computing resources to meet their instructional and administrative computing needs. In so doing, MECC seeks to achieve economies of scale in the purchase of computing hardware and communications facilities, cost avoidance in the common development of software systems, the benefits of interaction among levels of education and instructional disciplines, and the provision of computing access to all students within the educational system.

PLANNING

Each member of MECC is responsible for its own operational plan. For the seven regions this includes the installation of computing equipment under the master contract. The MECC staff works closely with the members' staffs in the development of these plans.

Beginning in February before each biennial session of the legislature, the members assess their needs and work interactively with the MECC staff to develop plans which fall within the master plan for the State. It is intended that the two staffs can agree on an operational plan for the next biennium before the member's staff takes its recommendations to its advisory committees. The members present their plans to their own advisory groups before they are officially transmitted to MECC. The MECC staff receives the plan as recommended by the member's advisory group. They review this version of the plan and make recommendations to the Facilities and Services Review Committee. This standing committee consists of a technically qualified peer group. After the review by the MECC staff and the Facilities Committee, the plan is transmitted with recommendation to the MECC Board of Directors for final resolution.

SERVICES

MECC has the responsibility for the development and implementation of application and systems software for a management information system to be installed by the seven elementary-secondary and vocational regions in the state. Applications are provided for the finance, personnel/payroll, student, and instructional management areas. The advantage to central development is that each school district will have access to comprehensive, standardized software for their improved management without the duplicative cost of developing such systems on its own.

As part of this effort, MECC has secured a master contract for a line of computing equipment to be installed in each of the regions. Through this bulk purchase approach a discount of approximately forty percent was achieved. It is anticipated that this contract will eventually result in a savings of approximately $80,000 per month over what could have been achieved through independent contracting.

Each of the regional groups are autonomous and thus, they are responsible for the delivery of their own MIS services once the software is installed. However, MECC will provide assistance in the installation and maintenance of the software.

MECC also has the responsibility of providing instructional timesharing services for all MECC member systems. To accomplish this, three systems are used. MECC operates a timesharing computer which provides 375 terminal accesses or ports to school district and higher education users. MECC also acts as a broker for services supplied by the University of Minnesota's MERITSS network. The TIES region supplies 96 interactive instructional ports. In all, approximately 1800 terminals are supported for public and private educational institutions throughout the state. Over 200 new instructional computer programs have been developed which are now shared by the user institutions.

In order to accomplish the services described, MECC has had to develop an extensive communications network which links the users to the computers. Purchase, installation and service of equipment are all services that MECC provides. It is estimated that the value of the communications network now stands at approximately one million dollars and that without MECC coordination and contracting, the equivalent cost of the network would be approximately $2.5 million. Bulk purchasing alone has resulted in a savings for the members of approximately 20 percent on the purchase of interactive typewriter terminals.

Provision of facilities is not the only service provided by MECC. In order for the educators and students to make effective use of the instructional timesharing system, the assistance of instructional coordinators who make up the field services team is required. Providing the documentation necessary to use the system is a major task. Conducting training sessions is another matter. Over 300 such workshops have been held each year. Not only is knowledge in the use of the instructional system gained through these workshops but also the ability to exchange ideas among faculty and staff from all levels and disciplines is offered.

The MECC staff also provides assistance in the special projects area. Assistance is provided to member systems that are interested in the exploration of innovative uses of the computer as an educational tool. Services include assistance in securing funds, project management and implementation. Special projects in the past have included interaction with a number of State and Federal agencies. MECC has worked with the State Department of Corrections to provide computer assisted educational materials. A project with the Minnesota Occupational Information System implemented a computer based package which matches a potential student's interests and aptitudes with vocations requiring these skills and provides information on educational institutions offering curricula in these areas. MECC has conducted a major study for the National Institute for Education on the impact of computers on school district administrative practices. A statewide study of the impact of instructional computing on computer literacy is being conducted for the National Science Foundation.

The consortium has made significant progress in the four years of its existence. It has achieved savings through bulk purchasing and the implementation of master contracts for computing equipment. But even more important is the expansion of services to those institutions and students who probably would not have received them had it not been for the formation of a statewide service organization. The fact that the percentage of school districts with access to instructional computing has grown in four years from seventeen percent to over seventy percent attests to the impact that MECC is having in Minnesota.

FACTORS ASSURING AN EVOLUTIONARY ROLE FOR THE COMPUTER IN CURRICULUM

Wallace Judd
IMSSS, Ventura Hall
Stanford University

For the computer to realize its full potential in educational use, it must become part of an evolutionary system. This paper presents considerations of software security, standardized student-record formats, and evaluation which promote evolutionary development. An example of each is presented.

Human classroom teachers currently present a limit to the complexity of curriculum, and limit the rate technological change can be applied in the classroom. Ramifications of the changes computers will cause in both areas are discussed.

Introduction

The present paper formulates a framework for the involvement of industry, government, and the schools in the application of computer technology to the issue of education. Without such a framework the fragmented efforts of all three principals will surely be much less effective than with a framework coordinating their efforts. The considerations are theoretical -- but the results have practical ramifications for the computer industry and for government involvement.

Unmediated Student-Author Contact

The time was September, 1901. The place was a little bicycle shop in Dayton, Ohio. Orvyl and Wilbur Wright had been working for several years on their gliders in an attempt to find the most efficient lifting surface shape. The problem they consistently faced was that there was no consistency in the wind conditions they encountered. Trials on one day would have cross-winds, on other days the winds were sporadic. As the winds varied, so did the lifting properties of various wing cross-sections. How could they find any uniformity, any general principles of lifting bodies, without constant wind? Wilbur finally hit on the solution! Why not make a wind? One that could be controlled by the experimenter. Though the wind tunnel had been discovered 20 years to Wilbur's re-invention, Wilbur alone realized that it was the key to manned flight. Several days later, Wilbur sent a letter to Octave Chanute, one of the era's foremost authorities on manned flight, that a practical, powered airplane "was now no more than a few years away." The wind tunnel meant that the Wrights could generate controlled conditions in which the only variations were the wind speeds, and the shape of the lifting body. Every other fluctuation in conditions was ruled out. Educational research is today in need of a "wind tunnel", a method of ruling out the infinite changes in condition due to teacher, classmates, and a myriad of other influences on the student. The computer, used either at home or in the school, will become such a "wind tunnel."

From the point of view of a curriculum designer, I have become very frustrated with pilot studies of my materials. One teacher will do an excellent job of presentation, expanding the materials where students need further involvement, adapting the games or activities where they are slightly difficult or easy, and motivating the students so they are genuinely interested in the subject presented. Other teachers may use the materials as a "reward", to motivate their use. In such a situation, kids who rarely get their "reward" may generate a 'sour grapes' response, putting down the materials and the students who are involved in them. Some teachers themselves feel negatively about the subject, and transfer their feelings to the students. The net effect of this teacher variable is to totally obscure the effect of the materials on a student, or on a class. As a consequence, learning theories cannot be applied to curriculum construction and be reasonably expected to show results. The teacher, classroom, and other environmental variables are so powerful that they overwhelm the experimental effects.

The computer allows the direct interaction between the designer and the student. This direct contact will allow curricula to be tested on their own merits, rather than as they are presented by the teacher. By providing a situation which substantially reduces extraneous variables in the learning process, the computer will be a "wind tunnel" which allows researchers to measure much more directly the effects of a learning theory and/or a specific curriculum. In its role as a teacher, the computer will provide for the evolution and sophistication of learning theory which was previously impossible.

Software Security

The context in which curriculum is commercially developed in this country is largely capitalistic: an author produces materials, and if they sell, the author receives large royalties; if they don't sell, the author receives small royalties, and likely drops out of the process. Very few educational authors write for the sheer pleasure of it. Even authors on government projects have the possibility of having their material sold. This capitalistic process is the foundation for a potentially evolutionary system.

A company cannot afford to fund a development project if it can't protect that project and realize a profit from it. That's what copyright was initially set up to do -- for print media. In the days before Xerox, the copyright law worked relatively well. It also worked moderately well for film and strip film. It works poorly on cassette tapes, and before the 1978 Copyright law, worked ineffectively on non-print medium.

A look at the principles involved in these effects shows under what conditions a copyright law can be enforced. A copyright law becomes more difficult to enforce when reproduction equipment and materials are cheap, and the product is expensive. Films are rarely duplicated because teachers don't have the reproduction equipment. The cassette duplicator, however, makes tape copying very attractive, and, incidentally, rather widespread.

To analyze the situation mathematically, the motivation toward copyright crime can be construed to be comprised of the following three elements: The retail price of the material to be copied; the cost of duplication, including raw materials as well as the copier; and the likelihood of getting caught and the punishment. Mathematically:

(Eq. 1) MOTIVATION = RETAIL PRICE -
(MATERIALS + DUPLICATING COST)
- (PROBABILITY OF GETTING CAUGHT
X PUNISHMENT) .

Or, briefly, the equation could be summarized:

(Eq. 2) MOTIVATION = RETAIL PRICE -
DUPLICATING COST - PUNISHMENT.

For Xeroxing, the temptation was not great initially, since the Duplicating Cost was high, relative to the Retail Price of the book or article to be copied. When copiers came into widespread availability, the Motivation became substantially higher, since the Duplicating Cost no longer had to be amortized over an individual person's copies.

For cassette tape, copies are made freely in district offices and in media centers. Cassettes are often duplicated (illegally) by districts, because the reproduction equipment is inexpensive and blank cassettes are about one tenth the cost of the retail educational tape. With a $250.00 tape copier, blank tapes at about $1.00 and retail tapes at $10.00, it takes copies of 28 tapes to recoup the initial investment in a copier. All copies over 28 save $9.00 per copied tape. A tape copier can pay for itself easily in the first six hours of operation. These facts make illicit copying very attractive. Simultaneously, the punishment is insignificant. Since the person copying has not sold copies of the tape, but simply distributed them, the extent of liability is limited to the revenue lost by the publisher and distributor. Few publishers will risk alienating customers for the meager satisfaction of a trivial court victory. As a result, the attempts that have been made to curtail cassette copying have focused on making tape cases physicaly impossible to fit into the copying machinery.

While it may seem that the discussion has gone a long way from computing, we can now see the ramifications for the above analysis for computer software development. A look at Equation 1 shows the potential for trouble. Computer software will have a very high Retail Cost; will have very low Raw Materials and Duplicating Cost; and despite the 1978 Copyright Law, will still have a low likelihood of Punishment. The situation will result in extremely high motivation for illicit copying. A look around most any educational computing conference will verify this fact even at the present time, for there are bulletin boards full of notices of people willing to 'trade' one program for another. Many of these are personally designed programs, but one does wonder if the same thing wouldn't happen with retail programs.

Clearly, the publisher (or whatever becomes the large source of commercial programs) has to amortize an extensive development cost in the Price of the cassette or disk. The bootlegger has no development cost to amortize -- just the price of copying, which is very low. The factors outlined above point to disaster not only for the publisher, but for the author of commercial programs as well. If extensive copying occurs, there will be little reason to develop intricate, well-reasoned software. The software development will be sponsored by the hardware manufacturers so they can sell hardware. Even in this function, its value will be dubious, since other manufacturers could put out machines on which another manufacturers' software can be run.

The solution has to be a software protect system, one which prevents use of a program on a machine it was not designed to be run on. This can be achieved readily by coding a "key" and a large composite integer into a ROM built into a machine. The key is one of two factors of the large composite number, the other factor of which is be on the leader of the program. As the program is read into the machine, it checks the ROM key, and uses the large, composite number to unscramble the microcoded program. An order for a program would include the "key", but not the composite number. The software produced as a result of the order would be good only for that machine, and for no other.

While this solution to the security problem is certainly able to be broken, it would require extensive effort. That extensive effort would result in a high initial cost for the bootlegger, so that his price for the bootlegged programs would also have to be rather high. Additionally, the legitimate manufacturer could check the key submitted for each order against a master list of produced keys. If the key is on a machine it was not intended for, then the illegal situation would be obvious -- and its location identified by the address on the purchase order. More importantly, the effort put into unscrambling the program and deciphering the ROM code constitutes intent to defraud on the part of the bootlegger -- a felony, rather than a misdemeanor violation of the copyright law.

In essence, what the composite integer "key" does is to raise the COST, as well as increase the PUNISHMENT in Equation 2. This will likely be sufficient to reduce illicit copying dramatically, and to make sure that software authors and producers are rewarded for their efforts.

Standard Student-Record Format

Without a standardized student-record format for storing information about a student in RAM, each program will have to work up its own history on a student. This, both from a curriculum and from an information storage point of view, is inefficient. For example, it seems reasonable that most programs would like to know the student's name. Should each one interrogate the student for that information? While that may seem trivial, it isn't when you also consider that many programs may want to ask the student his IQ, or his N-Ach, or perhaps his dominant modality. These are determined by extensive testing. To take advantage of these measures, must each program do the testing required for measures to which it is sensitive? If each program did indeed retest on the same dimension, a test practice effect is sure to arise. Clearly, there is a more reasonable solution.

Each student should be supplied with a RAM card, a cassette or a disk, which can be used as a transportable student record. This student record contains information formatted in a standard way, which any program can read. This would allow programs dramatic sensitivity to individual student characteristics, without having to invest all the time and energy of retesting for data. There could be a number of formats to choose from, depending on the information likely to be needed and the level of the student. Figure **1** shows example of what a format might contain, and a blocking design. The horizontal and vertical lines indicate block markers and word extents similar to conventional tape formatting markers.

Such a format would allow any program to make use of information stored in it. Although this format doesn't have the flexibility of storing unanticipated dimensions, since it doesn't have any blank words of storage, other formats which have room for new dimensions can be developed. Choice of the measures to include could be made by determining which measures have the highest weighting in factor analysis of the results. Alternatively, the included measures might come from the judgment of curriculum designers and psychologists as to which measures are most necessary to consider in the design of curriculum. Whatever basis is used for the choice of measures to be included in the format, the resulting data will be much more useful than having each program recollect and store the data it needs.

Compiler Standardization

A brief look at historical software developments reveals that standardization of the medium is a must. Audio records were perhaps the simplest to standardize initially. Since Edison was so dominant in the development of both the wax cylinder and the flat disc record, his formats emerged as the industry standard from the beginning. Only long after Edison's death did technological developments in the recording industry force record companies to experiment with 33 1/3, 45, and 16 2/3 rpm records. During the late 40's and early '50's, the industry was in flux, and it wasn't until

```
---------------------------------------------
||| FORMAT:                                ||
---------------------------------------------
|| NAME: FIRST MIDDLE LAST
TEACHER: MS, MISS, MR, MRS FIRST LAST
GRADE:
SCHOOL:
DISTRICT                                   |
---------------------------------------------
|| SEX:
MOTHER'S NAME: FIRST, LAST
ADDRESS:
NUMBER OF SIBLINGS:                        |
---------------------------------------------
|| IQ:        | IQ MEASURE:
MODALITY: TACTILE:   |VISUAL:   |AUDITORY:
        MEASURE:
N-ACH:    | N-ACH MEASURE:
DIFFERENTIATION:   |DIFFERENTIATION MEASURE:  |
---------------------------------------------
|| MATH LEVEL:
ADDITION: T1, date, measure, treatment:
T2, date, measure, treatment  |
T3, date, measure, treatment  |
     etc.
SUBTRACTION:
MULTIPLICATION:
DIVISION:
GEOMETRY:
PROBLEM-SOLVING:                           |
---------------------------------------------
|| INTERESTS:
        SPORTS: sport | weighting; sport | weig
        READING:
        MATH:
        MUSIC:
        SCOUTS:
        T.V. SHOWS:
        HOBBIES:                           |
---------------------------------------------
```

Figure 1 SAMPLE STANDARDIZED STUDENT-RECORD

the development of stereo in the late '50's that the stereo 33 1/3 rpm disc became the industry standard.

A similar development took place with tape cassettes. They were little used when initially developed in the mid-'40's. It wasn't until Norelco developed their cassette and dominated the market in 1954 that other software and hardware manufacturers standardized their tapes and players with the Norelco cassette. At that time, the industry burgeoned because tapes were interchangeable, and a purchaser could be sure that RCA's tape cassette would fit on their Macy's player.

Likewise, in the computer industry, it will be virtually impossible for a manufacturer to produce the variety of programs currently necessary to fit on the myriad compilers on the market. Even if a program is done in BASIC, there are enough versions that if anything sophisticated is done, it requires readaptation for each compiler. In addition, space and time advantages accrue to versions done in machine language -- although this is substantially less significant when you can devote a single system to a single learner, as is currently possible. The software security system outlined above, along with the standardized student record formats, must be adapted on a relatively broad scale to allow widespread, common purchase of similarly formatted discs, cassettes, or memory modules. Even with a machine translator from one language to another, the pletnora of machine languages could greatly increase the investment required to even begin creating computer curriculum for the schools. The greater the initial investment, the smaller the number of companies who could get into the field -- and the fewer authors, editors, and planners who could initiate commercially available materials.

Government-Sponsored Evaluation

Evaluation of even a six-week course presented by computer is virtually impossible for a local school district to do. The impossibility is compounded in courses of longer extent. This is made apparent using an evaluation-time ratio. The evaluation-time ratio is simply:

(Eq. 3) EVALUATION-TIME RATIO =
EVALUATION TIME / ACTIVITY TIME .

As will become evident in the following analysis, this ratio is a productive way to look at problems in curriculum evaluation. Even though the measures reviewed are accurate only in orders of magnitude, the outcome is the same. Precise data would add little to the argument.

Anyone who has ever evaluated textbooks for adoption by a school district will long remember those hours spent pouring over the piles of texts. But teachers grow proficient at the thumb test, and can evaluate a single year's math textbook in several hours. In general, the evaluation-time ratio is roughly 3/180, or 1/60 . Thus, to review 5 books, each covering 180 hours of instruction, one teacher would take 15 hours.

Review of films and filmstrips is more time-consuming. A reviewer must take at least one hour for each hour of film to be assessed. The reason is obvious -- unless you run the projector at high speed, you can't see the film in its entirety unless you sit through a showing. In this case, the evaluation-time ratio is 1/1.

When one looks at a computer program which has a rather large number of branches, the time necessary to review the adequacy of each combination of possible routes grows exponentially. Even if we assume the relatively conservative figure of 10 hours' evaluation for each hour of student activity, it would reasonably take 1800 hours to review 180 classroom hours. Here the ratio is 10/1, and the task of fairly evaluating a program would be impossible for an individual (teaching) teacher to undertake. Figure 2 shows the progression in level as the evaluation-time ratio increases. From the figure it is obvious that only state or possibly

MEDIUM	E/T RATIO	CURRENT EVALUATION LEVEL
Print	1/60	Individual Teacher / District
Film	1/1	District
Standardized Test	2/1	District / State
Computer Program	10/1	? ?

Figure 2. EVALUATION-TIME RATIO AND EVALUATION LEVEL

federal governments can evaluate computer curriculum more than a few hours long.

In the above discussion, the term computer program refers to a program which is not simply a linear teaching device, or a program which is basically linear but which can branch to two or three levels, then return to the main line; rather computer program' refers to a course which begins to fully utilize the flexibility of the computer. The program must be able to reinforce a lesson at a number of levels, with a number of different examples. It should be able to pace at a minimum of five or six levels, and should constantly monitor student progress and adjust its own pacing appropriately. These criteria, coupled with a sensitivity to the mode of presentation and to other student characteristics, make 10/1 seem like a very conservative figure.

If the government is to evaluate, how can it do so efficiently? If standardized student records (suggested above) are available for all students in a given population, then the student records can be collected after a given period of time, and the data evaluated for effectiveness of a given curriculum. For this to be possible, there have to be common breakpoints in the curriculum at which evaluations are made. Without common breakpoints, there is no way to compare one curriculum with another in any but an overall way. For example, if one single-year math program teaches fractions more effectively than another, but the other program teaches decimals more efficiently, then the summary results of the eighth grade might show them to be pretty much the same. But if fractions and decimals are evaluated at the same point, and with parallel instruments, then the evaluator can see which program is more effective for each segment of instruction. Again standardization is the key element: standardization of student record format, of instructional segments, and of evaluation instruments.

Note that the standardization is not of instructional programs. The standardization is a framework within which instructional materials may compete -- and be extensively, efficiently, and impartially evaluated. The evolution of instructional programs can only take place within such a framework.

Teacher Limitations on Curriculum

Imagine, if you will, a teacher who tries to make the materials appropriate for each student in terms of pacing, modality, reading level, the interests of the student, and motivation. And suppose that our imaginary teacher only tries to assess each factor at three levels, high (or fast), medium, and low (or slow). Our teacher would have to have 625 sets of material available for each subject taught. For four basic subjects in elementary school, this totals 1875 different instructional packages -- at a single grade level. Quite obviously, it would be impossible for a teacher to keep track of the physical materials, let alone a student's progress through them. The point is that the teacher is the limitation on the complexity of the curriculum. Simply because the teacher is human. A computer, on the other hand, can easily keep track of students in such a small array as a 5 x 5 x 5 x 5 . Indeed, the computer could just as easily incorporate 8 factors at as many as 8 levels, although the problem of constructing this many sets might tax the financial capability of most curriculum development companies.

Now, since most curriculum research has been conducted by human teachers, and since no curriculum research the author is familiar with (including computer research) has attempted to incorporate more than three psychological descriptors simultaneously, it is possible that our inability to dramatically influence human learning is due to the complexity of the human organism. If there are 20 important factors in human behavior, and our experimentation only manipulates three of those factors, the other factors overwhelm those manipulated by the experiment. It is much more feasible that when we can match eight or ten factors appropriately, we will be able to demonstrate various hypotheses of learning theory. Once the field gets consistent, replicateable results, it will have a foundation upon which to build a model of human learning. Until then, the models and experiments do not have the power to describe or exhibit the complexities of human behavior.

Finally, the evolution of technology in our society is much, much faster than the schools incorporate that technology. This lag time has been due both to the slow production of textbooks, and to the inertia presented by teachers. Even in cases (such as SMSG) in which the texts came out relatively quickly, teachers' attitudes toward the new learning materials could not be changed, even over a ten-year period. Often, it has taken a turnover of an entire generation of teachers before an innovation is taught in the schools. If, however, the teaching of specific subject matters is taught by computer, the computer can be reprogrammed in a relatively short period of time, and students can immediately begin to learn the newest developments in a technology.

As times between generations of technology continue to be reduced, this technology-society lag becomes more and more important. Past civilizations have risen (and fallen) because of their ability to adopt speedily new technologies such as gunpowder, steel, and sails. Our civilization's very place in the world may in an important way hinge on the effectiveness of computer curriculum, and the rate at which it can allow our populace to adopt technology.

METRO-II: AN EDUCATIONAL COMPUTER CONSORTIUM NEAR COMMENCEMENT

D. Thomas King
St. Paul Public Schools
St. Paul, MN 55102
Betse Gregor
METRO-II, Minneapolis, MN 55413

ABSTRACT:
Two years ago a report was presented concerning some of the early challenges and problems faced by the METRO-II computer consortium and its six Twin City area school districts. On July 1, 1978 METRO -II will begin providing a complete array of computerized management information services encompassing the areas of student systems, financial, personnel/payroll and related services. Utilizing a Burroughs host/satellite computer configuration the new system will be able to deliver expanded services in many applications areas. Expanded features include report-writer capability and on-line telecommunications, two services not readily available to member districts in the past.
This paper will serve as a status report--just prior to commencement. A survey of views, both positive and negative, as expressed by various member district personnel is presented and discussed. A more general perspective is given by each author--one representing a member school district, the other from the METRO-II consortium. A concluding statement including the advantages and disadvantages as perceived is provided. Great cost-savings was not found to be among the major advantages. More is being done with more expensive devices, systems and people. However, increased and more effective use of sophisticated educational computer services may well hold the greatest promise. Sharing creative efforts while pooling resources may prove to be the commanding justification for our efforts. Hindsight hopefully will prove much less unsettling if our foresight is deliberate and our commitment shared.

> "If anything can possibly go wrong, it surely will." -- Murphy's Law
>
> "If you do not expect the unexpected, you will not find it; for it is hard to be sought out, and difficult." -- Heraclitus

Introduction

"Commencement" was chosen judiciously in the title of this paper. In public schools, we expend as much as 13 years of diligent instructional efforts on our students, be they prodigies or prodded. This comprehensive, complex schooling process is designed to let our students begin--to commence their vocational plans. Although the number of students we serve has been decreasing rapidly, the complexity of related educational and organizational tasks has not. More information is demanded by more specialized personnel who want more reporting, more analysis, all of which requires more dollars and time. While costs are up, the school revenues based on student counts are down. The number of jobs available to poorly trained or mistrained students also is decreasing. We are faced with providing more extensive, expensive services. The sobering thought with which those of us in education are faced, simply stated, is: how can we do what we should do and what we must do, when we have less with which to do it? We, too, must commence to seek needed solutions. A consortium approach to educational computing may be a viable solution. The METRO-II computer consortium has been our collective approach toward these problems.

Those who have worked with educational MIS (Management Information Systems) constantly hear the litany of untenable data processing requests. You who labor in the electronic vineyards could help extend toward infinity the sequence of often near-impossible demands. Regardless, as professionals, we continue to do what we can with what we have. One suggested ameliorating or "bettering" solution has been to pool computing resources and creativities--to utilize a "consortium" approach.

Background

Nationally, nearly all of our larger school districts currently utilize computer services. A growing number of districts are studying the benefits of sharing resources and talents by forming an educational computer consortium. A consortium approach may have occurred by desire or by mandate. For almost all Minnesota school districts, political pressure and a legislative mandate came first. MECC (Minnesota Educational Computing Consortium) was formed as the statewide organization to coordinate all administrative and instructional computer applications in the State of Minnesota. Regionally, METRO-II began as a joint powers agreement to provide administrative computer services for each of six Twin City area school districts serving about 150,000 students: Minneapolis, St. Paul, Robbinsdale, Moundsview, District 916 AVTI, and South Washington County. METRO-II is one among seven such regions statewide, and was selected to pilot test the statewide systems developed by MECC.

Each of the METRO-II districts have had computerized information systems providing a variety of services to their district users. After July 1, 1978, however, the six school districts comprising METRO-II will share the planning, use, evaluation and costs of all administrative computer services. For hardware, METRO-II has adopted a Burroughs host/satellite configuration. Remote satellite computer equipment in each district will be linked via telephone lines to a large host computer at the central site. It is planned that video display terminals will be used extensively. This approach allows for some autonomous operations on smaller jobs, but brings cost savings on longer, more complex routines. The software is being developed by MECC under contract to the State Department of Education. As the pilot region, METRO-II continues to work closely with the MECC development team to test the new systems.

One of the authors has reported earlier ("Developing a Metro Computer Consortium: Promises and Problems," AEDS Proceedings, 1976) about the initial reasons and planning for METRO-II. This report will delineate briefly the current status of the various MIS systems as we approach operations commencement this summer. Further, some of the attitudes and concerns of selected district personnel involved in planning and operations at different levels are reported. Lastly, each author will provide a perspective from a district and consortium viewpoint, respectively.

Status Reports

1. FINANCE SYSTEM

The ESV-FIN (Elementary/Secondary/Vocational Technical-Finance) system was the first statewide system available for pilot testing at METRO-II. For those familiar with accounting procedures, the new system is a double entry, modified accrual basis accounting system designed in a modular fashion to allow each district to use only those capabilities it needs.

The finance pilot study began in the Robbinsdale school district in April 1977 following these procedures: at the end of a given accounting period, the Robbinsdale finance file was converted to the new ESV-FIN file structure. Various transactions were then sent against both systems. The "parallel" testing was maintained by programmatic transactions (Robbinsdale transactions converted to ESV-FIN transactions) and manually-entered transactions (double input by Robbinsdale staff). The use of the ESV-FIN system obviously meant considerable changes for the finance users. CRTs are being used extensively to enter transactions and corrections, to monitor batch activity, to control the processing, and to format and request reports. In July, 1977 at the beginning of the fiscal year, Robbinsdale went "live" on the new system.

As of March, 1978 two additional METRO-II districts are using the ESV-FIN system for production. In one of those districts, a critical input and print operation was converted to run with the ESV-FIN system. Conversion activities are well under way in the fourth district; the current parallel is projected to continue until fiscal year end, 1978. Conversion is scheduled for summer 1978 for the fifth and sixth METRO-II districts.

As the conversions continue and enhancements to the ESV-FIN system are made available, more features of the ESV-FIN system will be made available. For instance, St. Paul, the fourth district to convert, will begin using the accounts payable, accounts receivable and encumbering of requisition features. Inventory control, an enhancement, has been pilot tested in Robbinsdale during April, 1978. Property accounting and cost allocation will be available for the sixth district conversion.

2. PERSONNEL/PAYROLL SYSTEM

The Personnel/Payroll system currently is being tested in the Mounds View district. The parallel method of conversion described earlier is being used for this system conversion as well. Training activities, personnel reports and payroll generation, output control and verification are ongoing. The pilot is scheduled to end in May 1978, with full-scale payroll and personnel transactions operational in June.

Conversion planning is occurring in several other METRO-II districts. District review of the user documentation and system capability discussions is continuing in the other districts scheduled for conversion. These review activities are designed to identify, document and schedule the efforts required by each district to enable a programmatic data base conversion and parallel maintenance. Areas in the standard system which will need custom or unique modification also are being identified by district.

3. STUDENT SYSTEMS

The student systems are being piloted in the St. Paul district. To date, a test data base has been built and program testing activities begun in the student accounting, census, mark reporting and attendance/enrollment system areas. Current processing is in batch mode until an on-line generalized message control system can be implemented.

A separate Burroughs-supported system is being utilized at the present time for student scheduling. There are three METRO-II districts using this schedule, with another METRO-II district parallel testing one school on a newer version.

Discussions are continuing with METRO-II school districts to identify various system needs for transportation, special education, food service, history and other related areas of student systems. Programs to convert district files, help maintain the parallel processing, and convert unique student system programs currently operational in a particular district are being written and tested. Conversion availability for METRO-II districts currently is scheduled for fall 1978.

4. INSTRUCTIONAL MANAGEMENT

In December 1976 the design requirements for the Instructional Management Support system were released for statewide review. The design specifications included such areas as instructional activity, group and environment definition, instructional objective bank creation, student enrollment and transfer, test form, test scoring and achievement status.

Progress to date has centered on the creation of activity and objective banks and update program generation and testing. Report, edit and control programs also are being written. METRO-II soon will choose a pilot district to begin testing this system. Closely associated with this pilot activity will be the standardized and criterion-referenced test scoring systems.

5. RELATED METRO-II ACTIVITIES

In February 1978, METRO-II accepted from bid a Burroughs 6800 system to serve as the regional host computer. The various operating, software and application systems were gradually transferred from the statewide MECC system to the B6800. Finance production was transferred as each district closed its monthly accounting cycles. The communication network of remote job entry computers and cathode ray tubes is being developed. Current activity includes the ordering of computer satellite equipment for each METRO-II district. Policy formulation covers such critical areas as security of district files, authorization for information release, and a cost-charging procedure which will use differing factors for cost allocation as conversions continue.

Survey

One strong and essential feature of the cooperation between METRO-II and its six member districts has been the open and continued communication and dialogue. (Some have commented that at times, the dialogue gets too honest!) There must be an element of solving "our" problems; "their" problems are "our" problems.

Several years of effort involving hundreds of people has already been invested in the planning and development of METRO-II. As we approach our July 1, 1978 operations commencement date, in addition to a guarded optimism, there still are concerns. Some concerns have been minor, others more unsettling. One of the member districts, at this writing, has expressed deep concern about both hardware and software suitability.

To better assess the views and concerns of the METRO-II member districts, a short survey was prepared and sent to about 25 selected personnel throughout all six districts. Respondents were chosen as representative of all levels of school management. The questions presented are shown below with a synopsis of the many responses.

Question #1: Briefly, what was your reaction to the original decision to form METRO-II as a consortium for computer services?

Response Summary:

- theoretically a good idea, but difficult to put into operation
- since it was mandated, let's get going
- expect a loss of the effective, local computer systems now in operation
- better services at less cost should be a result
- the most effective method for meeting state reporting requirements
- should provide more flexibility in system capability and access to on-line systems
- expect a loss of control with greater turnaround
- working better than expected

Question #2: Briefly, what do you perceive as the two or three most difficult problems that your district has faced since METRO-II was formed?

Response Summary:

- trying to get people in various districts to work together and cooperate
- finding the best relationship between local and regional activity and authority; where should what be done
- the large time committment necessary from district staff
- the fear of change and resistance among district staff
- having to accept on faith that the new systems will work and will be acceptable substitutes for current local systems
- the fact that new developments at the local level are at a standstill
- METRO-II meeting production schedules
- the impossible timelines
- demonstrating that a local need is important enough to have development priority; "quick fixes" during production
- lack of good, trained staff in operations and systems development; problem of retaining competent staff
- lack of good documentation
- inadequate testing before pilot
- unreliable hardware and software
- the political pressure from outside sources such as legislature, state department

Question #3: Briefly, what do you anticipate as the two or three most difficult problems that your district will face after METRO-II becomes operational on July 1.

- training district staff, encouraging positive attitudes during conversions
- determining where priorities should be set; what those priorities should be
- handling the increase in user expectation because of promises and such actual system capabilities as on-line inquiry
- the sharing of increasing costs
- the ability to react with a limited staff to the immediacy of problems
- having to settle for standardized output
- learning to cooperate
- the degradation of services
- the reliability of hardware and software
- overcoming staff fear and resistance
- finding the best relationship between local and regional activity
- outside pressure from political sources

Other Comments

METRO-II should provide:

- improved documentation
- improved security
- higher level management involvement and information utilization
- more consistent lobbying for funding
- the stimulation for users to take advantage of new systems

A District View:

One author has participated as a school district representative in the development of METRO-II since its inception. We have come a long way since then, perhaps not always as quickly nor as directly as desired. Back in 1972, we had the largest K-12 data processing center in a five state area. Initially, then, our enthusiasm was guarded and our participation somewhat minimal (some might have said "downright negative"). As time passed, it became clear to us that participation would be prudent. Some philosopher must have written, "It is better to enjoy that which one must endure!" While "enjoy" may not be the correct word, we did become involved: we wanted to protect our interests, lose as little as possible, and attempt to direct the consortium as much as possible toward our needs. Within the St. Paul School District we set up two types of internal committees concerned with MIS: (1) an advisory committee (comprised primarily of major department heads) and (2) several user committees (student, finance, personnel/payroll, instruction and facilities/other). These committees have served to provide a liaison with METRO-II during planning and development purposes. Equally important, many members of these committees have gained better understandings of the extent and variety of services already available to them. These understandings often brought about more effective use of and requests for information by school people who previously had little or no experience with these systems. It appears that our participation in the METRO-II consortium has produced a more knowledgeable, sophisticated group of computer services users in our district.

There still is concern about loss of local computer services control. When jobs need to be done "yesterday", will we be able to get them finished before next week? In preparation for the new system, we hope to anticipate better, to avoid "crisis" operation wherever possible. It will be essential to continue to involve ourselves actively within the various user and advisory committees as well as our METRO-II Board activities. Problems of equitable cost-sharing are still being debated and are unsettling for some. Districts who considered themselves more advanced in certain applications (e.g. comprehensive student record histories, test profile analyses, research) are concerned that no degradation be suffered. Unique services cost more, however, and making such service available on the new larger system may be more expensive to a district than it would have been for them previously. Access to teleprocessing and report-writer type systems appears to be more readily available through the consortium than otherwise. Each district must purchase its own terminals and modems. Lower prices are made available through statewide bids. Also, the state has agreed to underwrite some of the communication costs. Even though all our METRO-II districts are relatively near the computer site, there should be considerable savings realized here as telecommunications operations grow.

One of the original rationales for the consortium approach was cost-savings. But although some costs may be higher, more is being done. Perhaps as a group, we are in a stronger position to face a mutual concern: what the State giveth, the State may taketh away. As a region, we can often speak with a single, louder, more influential voice.

Although our district has been a frequent and open critic during the various phases of METRO-II development, we believe that our criticisms and assistance have been constructive. We believe the consortium approach can work. We must make it work.

A METRO-II View :

The service staff at METRO-II has a unique vantage point. We are responsible for understanding and translating the concerns and needs of district system users to system developers. In turn, we are responsible for translating system capabilities to district users in the form of training, conversion coordination and trouble-shooting. The service staff is involved with most reported problems, whether they are related to a system problem or to an operational procedure.

METRO-II did not begin as a voluntary consortium. Each member district already had its own computer system providing services to its district staff. The districts knew and respected their data processing staff; that staff had given them timely service and knew the pressures associated with school administration. In contrast, the concept, the systems and the people of METRO-II are new and unknown. We at METRO-II know we must prove ourselves.

Because the conversion to METRO-II involves not only the testing of new application systems, but also new hardware and new vendor system software, many of the problems encountered to date are not surprising. While the solution to a specific problem is essential, the most detrimental aspect is that the sum total of the problems erode the confidence level of supported district staff.

From the beginning METRO-II has had the vital support of member district top management. The METRO-II Board of Directors realized that considerable extra effort and support was needed if the cooperative was going to work successfully. We are now at a point when the confidence and accompanying extra effort must spread through the entire consortium structure. We are at the point when district satellite computer equipment has been analyzed, approved by district school boards, and orders placed. The dates set for removal of current district computer equipment are now only months away. We must help ease some of the "trauma" being experienced as districts give up their current equipment and systems. We see this being done in two ways.

First, we must respond quickly and thoroughly to our defined problem areas. Continued assistance must be sought from other agencies to solve their problems which impact our production. There must be a time period of at least several months in which production schedules are met and systems shown to run without major programmatic problems. Our people need to be judged as competent, responsive and concerned.

Second, METRO-II must secure and then help focus the involvement of district staff. The areas which need attention have been identified. It is against those areas that district and METRO-II staff resources should be placed. METRO-II staff can encourage district involvement by more actively providing system information and opportunity for system experience. Extra efforts should be made to use already existing project teams. The project and conversion teams need specific objectives and management responsibility which ignore the normal boundaries of specific districts and consortium staff.

Only with the full use of all the talents and resources available in our member districts can the METRO-II consortium succeed.

Summary and Conclusion

There are many ways to view participation in a computing consortium. A cynical observer might state that "It's best to participate if you don't have any choice anyway!" An optimist might offer that "Together many things can be accomplished which would be difficult or impossible alone." Perhaps, the realist would combine the above views, realizing that, in public education today, there is a continuing erosion of local control as well as a need to work cooperatively. Our courts, and federal and state agencies increasingly dictate tighter operating parameters. While certain aspects of these controls may be changed or modified, the day-to-day need to provide good administrative services is not. The support of teaching and learning--education, if you will, is our most important, our only product. Although we are highly bureaucratized and regulated in public education, still we are motivated by trying to do what is ultimately best for students. The extent to which we can maintain a high level of service while we reduce our administrative costs, the more able we are to provide quality education. Providing optimal, economical support services is the ultimate rationale for and hope of any computer consortium. Those of us in METRO-II see these services as the major components of our educational catalyst. The chemistry of quality education cannot occur without the needed catalyst of our collective efforts.

Bertrand Russell may have been wrong, at least as his quote applies to us: "People used to think that when a thing changes, it must be in a state of change, and when a thing moves, it is in a state of motion. This is now known to be a mistake." We are changing; we are moving. The most rapidly-acting, powerful catalyst in educational technology is us.

A COMPUTER SIMULATION OF NAVIGATING AN AIRCRAFT USING VOR

Dr. Roland L. Meade
Program Supervisor for Instructional Applications of the Computer
Washtenaw Intermediate School District, P.O. Box 1406, Ann Arbor, MI 48106

A computer program to simulate navigation of an aircraft using VOR has been developed. The program is available in BASIC for the HP3000 and 2000 computer systems. The paper covers the following: operation of the program and a description of its features, modifications of the program to accommodate airports and VOR frequencies found outside of southern Michigan, development of a program using techniques from VOR to simulate VFR (visual flight rules) navigation, current uses of the program and educational settings where the simulation might be useful. Finally, transferring the program to other computer systems is covered.

INTRODUCTION

VOR is an acronym for Very high frequency Omni-directional Radio. This radio is used by pilots to navigate an aircraft on cross-country flights. A needle, called the Course Direction Indicator (CDI), in the VOR display tells the pilot when he is flying on a course to or from, the station to which he has tuned. The CDI deviates left or right if the plane is off course.

The VOR station sends out a radio signal (see FIG. 1) for each degree of 360°. These are called radials. To use VOR the pilot selects the station he wishes to use from his charts. He then tunes the radio to the frequency of that station. Each station also sends a unique call sign in Morse Code that he hears in his earphone to verify that he is on the right station. He then uses the Omni Bearing Selector (OBS) knob to set in the course or radial he wishes to fly. He can turn his aircraft until the needle is centered, to fly the given course. When the needle is centered he is on the selected radial.

A TO-FROM indicator gives the position of the plane relative to the station. As one passes over the station the TO-From indicator goes to "OFF" and then to the opposite reading. When the indicator goes "Off", the plane should be over the VOR station. An "Off" will also be displayed if the plane is on a radial that is at right angles to the radial set in the OBS.

It is also possible to get bearings on two VOR stations, draw lines on a map from the station along each radial for those two bearings, and determine one's location from where the lines intersect.

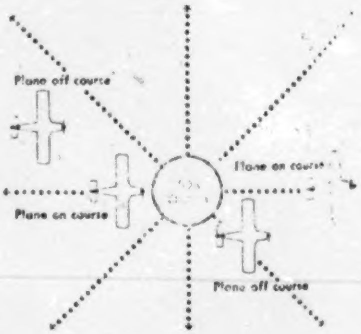


Fig. 1 - A VOR station sends out 360 radials, one for each degree on the compass. Eight radials are shown in the figure.

The Course Direction Indicator is independent of the aircraft heading. If the aircraft is on the radial set in the OBS, the needle will be centered. Normally, if the CDI deflects right, the course or radial is to the right and the aircraft should be turned right to that course. If the aircraft is flying on a heading that is opposite the course set on the OBS, the needle will still be centered. However, if the aircraft is in this position and the CDI is to the right, the course is to the left. This is called "reverse needle sensing." If the pilot wishes to avoid this condition, he should ensure that his course set on the OBS is the same, or within 90° of, the heading of the aircraft.

If the heading of the aircraft coincides with the course setting on the OBS, the TO-FROM indicator will read "TO" if the course will take the aircraft toward the station, and "FROM" if the course will take the aircraft away from the station. If the aircraft is headed 0° (north), is located north of the station, and the OBS is set at 180°, the indicator will read "TO". To determine this, if the aircraft heading were changed to 180° (the course setting of the OBS), it would be on a course TO the station.

THE VOR COMPUTER PROGRAM

All of the features mentioned above are contained in the VOR program. The program assumes the plane has taken off and is above the point of departure at an altitude of 9500 feet, travelling at 120 MPH, in the direction initially specified by the pilot. The following commands are available.

1.	+[N]	Changes the current heading by plus [N] degrees (E.G. +15)
2.	-[N]	Changes the current heading by minus [N] degrees (E.G. -5)
3.	C[N]	Sets the Omni-Bearing Selector knob to [N] (E.G. C175)
4.	D	Displays VOR readout
5.	F[N]	Sets the VOR frequency knob to [N] (E.G. F109.0)
6.	I	Types this list of commands
7.	L	Lands the aircraft
8.	Q[N]	Changes the time quantum by a factor [N] where [N] can take on values from 1 to 10 (E.G. Q10 - causes simulated time to be 10 times faster than real time)

The first two commands control the heading of the aircraft; the last is used to land. D displays the VOR readout. The display is also given if there is no input for thirty seconds.

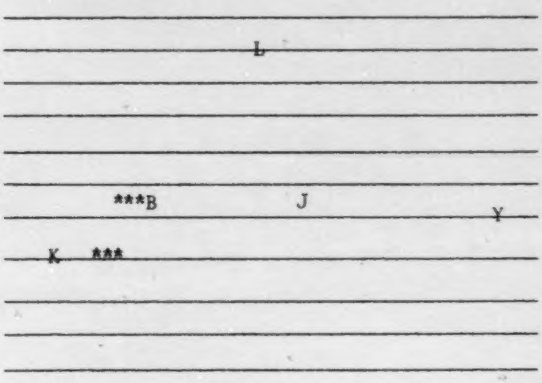
The C and F commands (3 and 5) operate the VOR unit. Commonly the pilot will set his course in the direction of the VOR station and then adjust his heading to fly directly toward the station. The needle will be centered when the aircraft is flying on the chosen radial. The simulated flight occurs in real time. This means the flight from Kalamazoo Michigan to Battle Creek will take about five minutes at the terminal. A flight from Kalamazoo to Willow Run, however, would take over an hour. By using the Q command real time can be compressed by a selected factor of 1 to 10 times. A sixty-minute flight can be compressed to six minutes. This feature is operative only if the user enters no commands and lets thirty seconds expire. If any commands are used, the pilot is returned to real time until he lets the program time out again. In other words, if he is flying with no changes, time is speeded up, if he uses any commands he is returned to real time to complete his corrections. The maximum flying time is the minimum amount of time it would take to fly between the airports chosen by the pilot, plus forty minutes. Forty minutes is the recommended minimum reserve for a cross-country flight. If you run out of gas, you will crash.

If you have flown a straight line from one airport to the other, you should be over your destination with forty minutes of fuel left. If you are over the VOR station, your "TO" indicator should change to "OFF". In any event you will eventually want to land using the L command. If you are within five miles of your destination you will be congratulated on your navigational skills. If you are farther than five miles, you may find yourself over an open field, or over wooded, hilly terrain. If you decide you are hopelessly lost, you can attempt to land with varying consequences. After landing the pilot can get a plot of his actual course. The pilot will find the wind will affect his course and he will have to make corrections during the flight.

A sample run follows:

```
VOR SIMULATION

For this simulation you will be flying a CESSNA
150 at an altitude of 9500 feet with an indica-
ted airspeed of 120 MPH.

The following airports may be used:

1.  Kalamazoo Municipal Airport
2.  Kellogg Field near Battle Creek
3.  Capital City Airport near Lansing
4.  Reynolds Field near Jackson
5.  Willow Run Airport near Ypsilanti

Type in the number of your point of departure,
the number of your destination, and your initial
heading on the same line separated by commas.
(E.G. for Willow Run to Kalamazoo Type 5,1,270.)

?1,2,80
You have been outbound from your point of depart-
ure for approximately five minutes.  Use VOR
navigation to complete your flight.  For a list
of the commands that may be used, Type 'I' after
the computer types 'Next?'

Remaining flight time:  45.4 Min.
NEXT?F109.4
Remaining flight time:  45.3 Min.
NEXT?C80
-... - .-..        -109.4-      ( 80)
L!*! ! ! ! ! ! !R   ( TO )
HEADING:  80 Degrees    Airspeed:  120 MPH
Remaining flight time:  44.9 Min.
NEXT?-10
New heading:  70 degrees.
Remaining flight time:  44.5 Min.
NEXT?C70
-... - .-..        109.4-       (70)
L! !*! ! ! ! ! !R   ( TO )
HEADING:  70 Degrees   Airspeed:   120 MPH
Remaining flight time:  44.3 Min.
NEXT?C65
-... - .-..        -109.4-      ( 65 )
L! ! !*! ! ! ! !R   ( TO )
HEADING:  70 Degrees   Airspeed:  120 MPH
Remaining flight time:  44.2 Min.
NEXT?C60
-... - .-..        -109.4-      ( 60 )
L! ! ! ! !*! ! !R   ( TO )
HEADING:  70 Degrees   Airspeed:   120 MPH
```

```
Remaining flight time:  44.0 Min.
NEXT?-8
New heading:  62 degrees.
Remaining flight time:  44.0 Min.
NEXT?C62
-... - .-..          -109.4-      ( 62 )
L! ! !*! ! ! ! !R   ( TO )
HEADING:  62 Degrees   Airspeed:     120 MPH
Remaining flight time:  43.9 Min.
NEXT?C61
-... - .-..          -109.4-      ( 61 )
L! ! ! !*! ! ! !R     ( TO)
HEADING:  62 Degrees   Airspeed:     120 MPH
Remaining flight time:  43.6 Min.
NEXT?-1
New heading:  61 degrees.
Remaining flight time:  43.6 Min.
NEXT?Q1_5
Remaining flight time:  43.5 Min.
NEXT?D
-... - .-..          -109.4-      ( 61 )
L! ! !*! ! ! ! !R       ( TO )
HEADING:  61 Degrees   Airspeed:     120 MPH
Remaining flight time:  43.1 Min.
NEXT?Q2
Remaining flight time:  43.0 Min.
NEXT? -1
New heading:  60 Degrees.
Remaining flight time:  42.8 Min.
NEXT?D
-... - .-..          -109.4-      ( 61 )
L! ! !*! ! ! ! !R                 ( TO )
HEADING:  60 Degrees   Airspeed:     120 MPH
Remaining flight time:  42.7 Min.
NEXT?C60
-... - .-..          -109.4-      ( 60 )
L! ! !*! ! ! ! !R       ( TO )
HEADING:  60 Degrees   Airspeed:     120 MPH
Remaining flight time:  42.5 Min.
NEXT?C58
-... - .-..          -109.4-      ( 58 )
L! ! ! !*! ! ! !R                 ( TO )
HEADING:  60 Degrees   Airspeed:     120 MPH
Remaining flight time:  42.4 Min.
NEXT?-3
New heading:  57 degrees.
Remaining flight time:  42.3 Min.
NEXT?D
-... - .-..          -109.4-      ( 58 )
L! ! ! !*! ! ! !R       ( TO )
HEADING:  57 Degrees   Airspeed:     120 MPH
Remaining flight time:  42.2 Min.
NEXT?
-... - .-..          -109.4-      ( 58 )
L! ! !*! ! ! ! !R    ( TO )
HEADING:  57 Degrees   Airspeed:     120 MPH
Remaining flight time:  41.1 Min.
NEXT?C57
-... - .-..          -109.4-      ( 57 )
L! ! !*! ! ! ! !R       ( TO )
HEADING:  57 Degrees   Airspeed:     120 MPH
Remaining flight time:  40.9 Min.
NEXT?C56
-... - .-..          -109.4-      ( 56 )
L! ! !*! ! ! ! !R    ( TO )
HEADING:  57 Degrees   Airspeed:  120 MPH
Remaining flight time:  40.8 Min.
NEXT?C54
-... - .-..          -109.4-      ( 54 )
L! !*! ! ! ! ! !R       ( TO )
HEADING:  57 Degrees     Airspeed:   120 MPH
Remaining flight time:  40.7 Min.
NEXT?C58
-... - .-..          -109.4-      (58)
L! !*! ! ! ! ! !R          ( TO )
HEADING:  57 Degrees    Airspeed:   120 MPH
Remaining flight time:  40.5 Min.
NEXT?-5
New heading:  52 degrees.
Remaining flight time:  40.3 Min.
NEXT?C52
-... - .-..          -109.4-      ( 52 )
L!*! ! ! ! ! ! !R      ( TO )
HEADING:  52 Degrees    Airspeed:   120 MPH
Remaining flight time:  40.1 Min.
NEXT?C60
-... - .-..          -109.4-      ( 60 )
L! ! ! !*! ! ! !R  (*OFF)
HEADING:  52 degrees    Airspeed:  120 MPH
Remaining flight time:  39.9 Min.
NEXT?L
Excellent job!  You made a perfect landing.  Do
you want to continue on to another destination
(1=Yes, 0=No)?0

Do you want your course plotted (1=Yes, 0=No)?1
- - - - - - - - - - - - - - - - - - - - - - - -
```



```
- - - - - - - - - - - - - - - - - - - - - - - -

DONE
```

PROGRAMMING CONSIDERATIONS

Table 1 gives the five airports currently in the simulation, the coordinates, the frequency of the VOR station and the Morse Code call sign of the station. The origin and units for the coordinate system were chosen arbitrarily. Airports for another geographical region could be substituted in the program by changing the data in the program. More airports and VOR stations may be added.

The HP2000 version, written in BASIC, has been written using subroutines to facilitate modification of the program. The "ENTER" statement is unique to the HP computer systems and will probably be a problem for anyone wanting to modify the program for use on other systems.

Stations and Coord.	VOR Freq. and Morse Code
Kalamazoo -20, 20	109.0 .- --.. ---
Kellogg -24,35.5	109.4 -... - .-..
Capital -57.5,68	110.8 .-.. .--.
Reynolds -22,77	109.6 .--- -..- -.
Willow Run -21,124	110.0 -.-- .. .--.

Table 1. Airports and VOR stations currently used in the program.

VFR

The techniques found in VOR should be useful in writing a program to simulate VFR (visual flight rules) flights. Landmarks along the flight path would be entered as data along with the coordinates of the landmarks. Instead of a Display command, one would type Look. All of the landmarks within a given distance of the aircraft would be printed out along with its bearing from the aircraft. The same basic code as used in VOR would be used to keep track of the aircraft's geographical location and its course. Airports would be landmarks and their coordinates entered as any other landmark.

Initially it would be easier to use a constant altitude. Later the altitude could be made a variable in the program. Climb and Dive would be commands that would change the altitude. If the aircraft descended below the minimum for VFR a warning would be printed; if it descended below the ground level it would crash. A grid could be set up, defined by the coordinates. The ground elevation within a grid would be the highest point within that grid.

The author began work on a VFR program last summer but was prevented from continuing due to the press of other duties.

Uses of Program

VOR is currently available to users of Washtenaw Intermediate Time Sharing in Washtenaw County, Michigan (includes the Ann Arbor area). It is also available to users of the HP3000 at Kalamazoo Public Schools. It has been used by individual students as they explore our central library. A teacher at Ypsilanti High School who is a certified flight instructor, has used it in his ground school classes to teach navigation using VOR.

Suggested Uses in Educational Settings

The program allows one to learn how to use the VOR unit without an aircraft, which is expensive and time-consuming to operate. If the student is familiar with VOR navigation, when he does actually begin in-flight instruction, he will have at least one piece of equipment with which he is familiar. The simulation also allows the user to "do" something in aviation rather than just read about it.

There are many ways in which the VOR program can be used. The simplest way is to determine the bearing of the station for a given airport, set the course for this station and then "home" on the station. In addition, the student can learn to set any course, not necessarily toward a station, and fly that course. This is useful if you want to fly to a place that is not the location of a VOR station. To do this one could locate his position on a map using two VOR stations. He would determine his bearing from a station by adjusting the OBS until the CDI is centered. After drawing in the radial from that station, he would repeat the procedure for a second station. His position would be the intersection of the two radials. All of these situations could be encountered in actual flight. Cecil Hebrew of Ypsilanti High School has suggested the following problem, that he uses with his students. The pilot takes off from Willow Run airport with a destination of Capital City airport near Lansing. He is to use the Lansing VOR and fly due north until he intercepts the 100° radial. He then turns and flys along the 100° radial until he is over the station. The station is located about six miles southwest of the airport, so the pilot must now turn to a heading of 60° and fly about six miles until he is over the airport. He should get the actual direction and distance from his aeronautical chart for Michigan. The pilot should try to fly as accurately as possible to keep his fuel consumption to a minimum. As the pilot develops his skills, he should try to land as close to the airport as possible, even though there is a five mile tolerance.

Obtaining the Program

Those with Hewlett-Packard 2000 or 3000 series systems can obtain a magnetic tape of the program. Users of other systems will probably need to obtain a paper tape or a listing of the program. The author does not have access to cassette tape equipment. Please contact the following for information.

Dr. Roland Meade
Washtenaw Intermediate School District
P. O. Box 1406
Ann Arbor, MI 48106 (313 769-6522)

The author wishes to acknowledge Dr. Garrett VanderLugt, Kalamazoo Public Schools, who wrote the original computer program for the HP3000.

Reference:
Private Pilot Manual,
Sanderson, 1976

A STUDY OF COMPUTER USE AND LITERACY IN SCHOOLS

A PROJECT PROGRESS REPORT

Daniel L. Klassen
Minnesota Educational Computing Consortium

The National Science Foundation through its Research in Science Education program has awarded a grant to the Minnesota Educational Computing Consortium's Special Projects Division to explore the impact of pre-college educational programs designed to increase computer literacy, and the effects of human-computer interaction within the context of instructional computing environments. The study will attempt to answer two basic questions (1) What is the impact on student knowledge, attitudes and skills of various approaches to the development of computer literacy? and (2) What impact does using a computer as part of the instructional process in science education have on student attitudes and beliefs about computers? In an effort to inform interested educators of the progress of the research effort and to gain reactions to projected assessment techniques, several members of the project research team will share information regarding current activities at the convention of the Association for Educational Data Systems.

One of the most fundamental goals of the American educational system has been the development of students who are literate. The concept of literacy when applied to reading, writing and arithmetic usually refers to a personal capability to acquire knowledge and skills essential for effectively functioning as a member of a community. When viewed from a broader perspective it is also a measure of a society's well-being and its political identity.

Recently educators have attempted to link the concept of literacy to the computer and its use and impact on society and the individual. The linkage has spawned the development of instructional programs, courses and units all of which attempt to create computer literacy. Many of these programs, courses and instructional units are now being implemented in schools ranging from elementary to the graduate level. They vary widely in their approach as well as in their content.

The project is divided into two major substudies. A field study will examine the impact variations in computer programming (science) and computers and society courses have on student knowledge, attitudes and skills related to computers. In addition to developing badly needed measurement tools, the study will produce empirical data regarding the impact/effectiveness of various approaches.

A laboratory study will be used to determine the impact of actual computer use on student attitudes toward computers. This effort will also produce measurement techniques and will begin to identify the role played by the computer and computer-based learning materials on student computer literacy. Participants in both studies will be drawn from schools in Minnesota.

Initial project activities have included the development of computer literacy assessment procedures, a survey of 7,000 teachers regarding the nature and scope of computer use in the classroom and the design of "computer experiences" to be used in the study.

COMPUTER-BASED INSTRUCTION:
NETWORK STATUS IN
NORTH CAROLINA

R. R. Pearson
N. C. Educational Computing Service

The North Carolina Educational Computing Service (NCECS) traces its origins to 1966 when it began as an instructional development project to acquaint institutions with the usefulness of computers in higher education. Today it operates as a supplier of services and support to institutions throughout North Carolina, not only from the Triangle Universities Computation Center (TUCC), but from computer centers and service bureaus throughout the United States. Its activities are user-oriented, providing an opportunity for teachers in all subject areas to preview, test, or review computer-based instructional materials from many sources prior to adoption and operational implementation.

The North Carolina Educational Computing Service (NCECS) traces its origins to 1966 when it began as an instructional development project to acquaint institutions with the usefulness of computers in higher education. It was funded by $300,000 from several private foundations, christened North Carolina Computer Orientation Project (NCCOP) and co-located with the Triangle Universities Computation Center (TUCC) in the Research Triangle Park between Durham, Raleigh, and Chapel Hill. TUCC is a not-for-profit corporation owned by Duke University, North Carolina State University, and The University of North Carolina at Chapel Hill. At that time (1966), TUCC operated an IBM S/360 Model 75 serving Model 30's at the owner universities. NCCOP became the "Fourth University", offering services to the other postsecondary educational institutions in North Carolina. TUCC has since expanded its hardware to dual IBM S/370 Model 165's and a Hewlett Packard 2000 mini-computer system, and NCECS has expanded to provide its users not only TUCC service, but also service from other host systems both in and out of North Carolina.

The original orientation concept was quite interesting. Each institution could make application for a year's free use of a Teletype (ASR 33), communication links, the Model 75, the rapidly-growing program library, and training workshops. At the end of the free trial, they could continue the service for a fee, drop it, or install their own machine. During the first few years 38 institutions participated in the program and the number of paying network members grew from 29 to a plateau of about 55.

In 1969 NCECS received a grant from the National Science Foundation (NSF) which supported an active and exemplary curriculum development project and in turn increased the size of the TUCC program library. A "package" of computer-based instructional materials would be identified and installed on the system, and then a group of potential users would be invited to a workshop to learn how to use it.

The activity with computer-based materials led to the founding and funding (by NSF) of CONDUIT in 1971 in an attempt to share (transport) materials of high quality between five regional networks. CONDUIT, located in Iowa City, Iowa, has become our principle source of new materials for the undergraduate curricula. Its services are available and used throughout the English-speaking world.

In 1972 NCECS became a program division of The (newly-reorganized) University of North Carolina which consists of 16 constituent institutions throughout the state. NCECS service continues to be available on an optional basis to these universities as well as the private institutions and community colleges in North Carolina. The University of North Carolina became a member of the Planning Council of EDUCOM in 1974 and these activities led to our present EDUNET project which links the TUCC facilities to TELENET and encourages our users to access host systems at universities in other states (and invites foreign users to access TUCC). This is a one-year "marketing" experiment and it has generated a good deal of interest. Our users are tending to use other collegiate centers for a) access to special resources, b) an economic advantage or c) to preview and evaluate a special resource in anticipation of importing it.

A year ago NCECS became a member of the Health Education Network (HEN) and began accessing extensive collections of computer-assisted instructional (CAI) materials for medical students and practitioners, as well as for students of nursing and allied health. These materials are produced and supplied by Massachusetts General Hospital in conjunction with The Harvard Medical School, and by The Ohio State University College of Medicine. Two additional hosts will be available shortly.

Access to and the development of statistical data bases has been an important part of the NCECS history, but only in the last year have we become involved with bibliographical data bases and access systems. We are just beginning to provide DIALOG capability to Schools of Library Science, and my feeling is that the library "market" will surpass most of the others. Learning materials--whether they are books, filmstrips, cassettes, or computer-based--belong in the library.

The model is a cascade of service centers. At the "head" end is the desired computing resource: some program, programming system or data base. Use of that resource on a national basis may be coordinated by an organization like EDUNET or HEN. The communications contracts are written with a packet-switching agent, like TYMNET, and access is made available to a regional distributor like NCECS. The distributor may contract with a computer center like TUCC to provide a service to a campus center, like the Academic Computing Center at The University of North Carolina at Greensboro (UNC-G) which in turn bills the teacher of chemistry who authorizes and assigns students to execute the simulation or "what-have-you". The model is working but the interfaces are not perfect.

This collection of resources and distributors provides the classroom teacher with swift access to materials that heretofore could have taken years to implement locally. The service is not inexpensive unless the dissemination rates, decision schedules and delay costs are included.

In view of the rapidly decreasing costs of local computing hardware and storage devices, this investment in expanded network service is justified by the belief that the most expensive and beneficial service to teachers is a properly-documented applications system and a faculty training activity which draws on the experience of successful adoptions. The strength of the network is the size, quality and variety of the library and the usefulness of its supporting documentation. NCECS offers a variety of training and sharing workshops in all subject areas. These meetings are typically a day and a half long, and conducted by staff members, borrowing heavily from faculty users. Teachers that have had successful experiences are asked to make presentations. New materials are examined and general discussion periods provide an opportunity for exchange of opinion. During this academic year, NCECS has scheduled faculty workshops on SPSS, biology, business, social sciences, mathematics and MINITAB, as well as technical workshops in TSO (our time-sharing system), Job Control, and special interest workshops for administrative users.

When a teacher begins to use these materials, a toll-free telephone consulting service is available to help with problems of any nature. This user service is also available for advice on new terminals or programs or a new user requirement. NCECS currently supports about 115 terminals. About 60 percent are interactive devices and the remainder are batch machines. NCECS also provides advice to planning agencies that are concerned with computers, communications and their appropriate uses. At the present time there are several planning activities underway. NCECS is considering new services in support of the handicapped, for word processing, libraries, career counseling and computer-assisted instruction for basic academic skills, some of which may be useful at the elementary and secondary school level.

In its role as a broker or distributor of educational computing service, NCECS has a responsibility to the user. Our activities are designed to provide the user with service to meet the needs of the campus. NCECS does not own a computer. We have no capital investment to protect. After about a dozen years in support of computer-based instruction, NCECS has evolved to its present position, emphasizing responsiveness to requests from the teacher and providing regular opportunities for faculty development. We anticipate that this model will be successful for some time to come.

REFERENCES

The following papers provide additional information on NCECS and the TUCC community:

1. "Organizational, Financial, and Political Aspects of a Three University Computing Center" by F. P. Brooks, J. K. Ferrell, and T. M. Gallie, Proceedings of the IFIP Congress, 1968, E49-52.
2. "Efficiency vs. Responsiveness in a Multiple-Service Computer Facility" by D. N. Freeman and R. R. Pearson, Proceedings of the 1968 ACM Annual Conference.
3. "Introducing Computing to Smaller Colleges and Universities - a Progress Report" by L. T. Parker, T. M. Gallie, F. P. Brooks, and J. K. Ferrell, Communications of ACM, Volume 12, 1969, pp. 319-323.
4. "A Functioning Computer Network for Higher Education in North Carolina" by L. H. Williams, AFIPS Conference Proceedings, Volume 41, 1972 Fall Joint Computer Conference, pp. 899-904.
5. "A Network Model for Delivering Computer Power and Curriculum Enhancement for Higher Education: The N. C. Educational Computing Service" by L. T. Parker, Jr. and J. R. Denk, EDUCOM Bulletin, Spring 1974, pp. 24-30.

AUTOMATION AND THE TEACHER EDUCATION PROGRAM

Thomas M. Roy
Southern Illinois University-Carbondale

This paper deals with the administration of a teacher education program, and specifically how we were able to use the computer to assist in this administration. Most educators are aware of the problems in the teacher's job market, at this time, in regard to supply and demand. For this reason, there is increasing pressure on teacher education programs to admit and retain only the better students. This task involves many administrative duties which were being performed manually, thus detracting from the time available for students. We have been able, through the use of Mark IV language, to eliminate many of these tasks and hope to eradicate even more in the near future. We were able to accomplish this without the services of professional programmers, who were busy maintaining the master file from which we obtained our student data. The use of a data retrieval language such as Mark IV allowed our current staff to accomplish this programming.

Introduction

Typically, a teacher education program has higher grade point requirements than other units of a college or university. The teaching profession is the means through which society receives its formal training. Thus, teacher preparation of the highest quality must be the first concern of the teaching profession. There is no way to insure that all prospective teachers will be good teachers, but a step in that direction can be taken by enrolling in the best teacher education program available. A good teacher education program has been described as one including general education, specialization, and professional education. The quality of a teacher education program is determined largely by its admission and retention policies and the strictness with which these policies are enforced.

Admission

Any student who plans to complete the teacher education program at Southern Illinois University at Carbondale must be admitted officially to that program. A student may apply to the Teacher Education Program with a minimum of 30 semester hours of academic work, completion of the introductory education course, (Education 201, Teachers Role in Public School Education), and an overall grade point average of at least 2.15 on a 4.00 scale. These criteria are common to the majority of teacher education programs in Illinois, as indicated in this excerpt from Profiles of Teacher Preparation Curricula in Illinois (p. 53). "When asked to check the criteria used for selection of prospective teachers from a given list, four were identified as used most often by all curricula: grade point average, minimum number of credit hours earned, formal application and personal interview. Formal application was the most frequently checked criterion. About 72.3 percent of all curricula had checked it... The second most frequently used criterion for selection of prospective teachers by all curricula was grade point average. Of all 220 identifed curricula, 70.9 percent had checked this criterion... Approximately the same percentage of all curricula reported minimum number of credit hours earned (52.7 percent) and personal interview (51.4 percent) as criteria for selection." In the past, calculation of grade point averages and hours of credit, has been a time consuming task for the advisement staff of the College of Education. These and other clerical tasks had detracted from the time which the advisement staff could spend with students.

In order to solve this problem, we have turned to the computer and Mark IV language. This language, marketed by Informatics Incorporated, is certainly not the only computer language capable of handling our tasks, but it is an excellent language for users without previous data processing experience. It is easy to code, and very efficient for a language of this type. We have produced computer printouts listing the student's name, identification number, grade point average, total hours, major and college. These listings have greatly reduced the time spent in verifying the criteria used for admission to our teacher education program. An extract of the Active Student Profile, which is the student master file for the University, is available for access by departments other than computing services, and this is the file which our Mark IV programs access for these listings. (See Figure 1). These programs are run several times during the semester in order to keep the lists as up to date as possible. The data changes due to incoming transcripts from other schools, and major changes, and grades changing from incomplete to a letter grade necessitate frequent runs.

Retention

Now that we have admitted the student to the teacher education program, our attention focuses on the retention criteria. The retention of students in teacher education is a hot item at this time. This is mainly due to the supply/demand ratio for teachers. According to Illinois Teacher Supply and Demand (p. 54), "The projected teachers supply/demand balance from 1976-77 through 1978-79 shows that the oversupply of teachers in Illinois is expected to continue, and that the gap between supply and demand is expected to widen considerably." It is for this reason that we must be sure that each student retained in the teacher education program is performing up to the standards set for this program. The most important criterion for retention and progress in the program is again the grade point average. The criterion is substantiated by Profiles (p. 64). "Evidence of a surplus of teachers during recent years as well as an increase in concern over the quality of education in elementary and secondary schools has caused an increased interest not only in criteria for selection of prospective teachers, but also in criteria for retention of prospective teachers. In an attempt to document the use of criteria for the retention of prospective teachers in teacher preparation curricula, respondents to this study were asked to identify those criteria used for retention purposes in their respective curricula. Of the ten choices given to respondents, only one was checked by over half of all curricula--minimum grade point average in all courses must be maintained. This criteria was checked by 89.1 percent of all curricula."

For these reasons, the next MARK IV system was written. This system consists of three programs, each producing multiple output formats. The first program selects students whose grade point average has fallen below the 2.15 average required for retention in the Teacher Education Program. A listing is produced of these students by name sequence within the student's major department. This list also shows the student's address, allowing Teacher Education Services to request the student to appear for a conference. Mailing labels for this letter are produced as a second output format. The final output format for the program is a 3 1/2" x 7" continuous form card which is placed in the student's advisement folder to alert the adviser of the problem. (See Figure 2.) The second program selects students enrolled in professional education courses without having been enrolled in the Teacher Education Program. The listing produced by this program is sequenced by course and section enabling Teacher Education Services to distribute these lists to the instructors. The continuous form card is produced for placement in the advisement file. For each student's card produced a trailing signature card is also produced allowing the adviser to notify Teacher Education Services of the action taken to enroll the student properly in the Teacher Education Program. (See Figure 3.) The third program monitors a special exception to the program which allows a transfer student with 60 semester hours or an Associate Degree to take a prerequisite course to the program concurrently with any of the program courses. If these restrictions are violated the continuous form card with signature trailer is produced and brought to the attention of the student and advisor. These listings must be produced during the first week of each semester in order to allow the student to change his schedule if necessary.

We have been able to produce many other listings for the advisement staff and Teacher Education Services office by using the Mark IV language. Lists of students approaching graduation have been very helpful in assuring that the graduation check is completed on these students. A follow-up questionnaire is printed on letter stock for recent graduates of our program for the department and the college. (See Figure 1)

Current and Future Projects

Our present computer project involves coding a term/year date of admission to the Teacher Education Program. These data are being added to the master file of student information and our extract of this file so that we will be able to produce statistical reports concerning admission and matriculation in the program. Our future plans include several more computer generated letters, such as an evaluation form for advisors, and the notices sent to the students selected by the retention monitoring system. These letters are now typed manually. A particularly exciting prospect for the future is the modification being studied now, which would make the student master file a longitudinal file containing a complete transcript of all current students. This would allow us to check prerequisites and design graduation checks for the computer.

In closing, I would like to mention two very important facts concerning our computer efforts in the Teacher Education Program. First, the computer systems have not replaced a single position among our staff. The advisors have had more time to spend with the students, and have been able to improve the quality of service provided for the students. Second, the concept of non-professionals in the data processing field providing a service such as this has worked well in our environment. I think it would work well in many situations, and is due in great part to the availability of a language such as Mark IV. This language is easily learned by the uninitiated for basic retrieval type applications, while, at the same time, it is sophisticated to the point of handling all the processing for a computer operation. This concept takes much of the small job pressure off of the professionals, thus allowing them more time for higher level endeavors.

Reference List

Illinois Office of Education. Illinois Teacher Supply and Demand 1975-76. Department of Planning and Research, Fall 1976.

Illinois Office of Education. Profiles of Teacher Preparation Curricula in Illinois. Department of Planning and Research, Fall 1976.

FIGURE 1

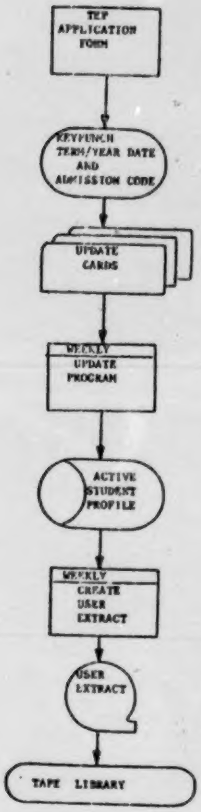

TEP APPLICATION FORM
KEYPUNCH TERM/YEAR DATE AND ADMISSION CODE
UPDATE CARDS
WEEKLY
UPDATE PROGRAM
ACTIVE STUDENT PROFILE
WEEKLY
CREATE USER EXTRACT
USER EXTRACT
TAPE LIBRARY

FIGURE 2

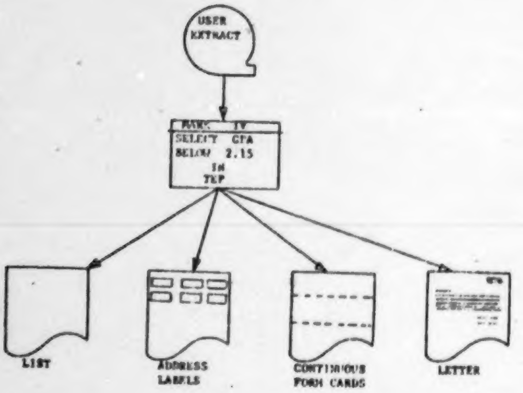

USER
EXTRACT
MARK IV
SELECT GPA
BELOW 2.15
IN
TEP
LIST
ADDRESS
LABELS
CONTINUOUS
FORM CARDS
LETTER

FIGURE 3

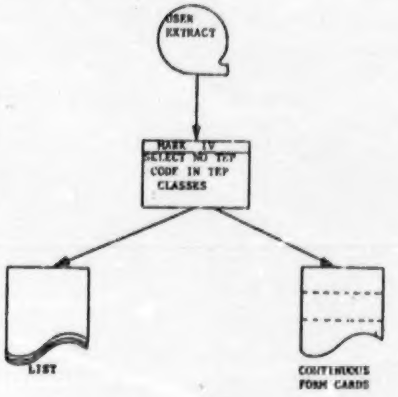

USER
EXTRACT
MARK IV
SELECT NO TEP
CODE IN TEP
CLASSES
LIST
CONTINUOUS
FORM CARDS

FIGURE 4

COLLEGE OF EDUCATION
TEACHER EDUCATION SERVICES
WHAM HALL 108
SOUTHERN ILLINOIS UNIVERSITY
CARBONDALE, ILLINOIS 62901

OCT 31, 1977

[redacted]
EDWARDSVILLE ILL 62025

DEAR GRADUATE:

WE HAVE BEEN REQUESTED BY THE COLLEGE OF EDUCATION, SPECIFICALLY THE DEPARTMENT OF CURRICULUM INSTRUCTION AND MEDIA, TO CONDUCT A FOLLOW-UP OF RECENT GRADUATES OF THIS COLLEGE.

THIS FOLLOW-UP HAS BEEN DESIGNED TO INVOLVE VERY LITTLE OF YOUR TIME, BUT PLEASE USE THE BACK OF THE FORM FOR ANY COMMENTS RELATIVE TO THE COLLEGE OF EDUCATION, YOUR MAJOR DEPARTMENT,OR YOUR EMPLOYMENT SITUATION.

PLEASE ANSWER THE FOLLOWING QUESTIONS:

1) WHAT COURSE OR COURSES DID YOU CONSIDER MOST BENEFICIAL?

____

2) WHAT COURSE OR COURSES DID YOU CONSIDER LEAST BENEFICIAL?

____

3) HAVE YOU OBTAINED EMPLOYMENT IN TEACHING OR OTHER AREAS?
IN SECONDARY EDUCATION ____________________
IN ELEMENTARY EDUCATION ____________________
SUBSTITUTE TEACHING ____________________
OTHER AREAS ____________________

4) IF EMPLOYED IN A TEACHING PROFESSION, DO YOU AGREE THAT YOUR PROFESSIONAL EDUCATION EXPERIENCE (EDUC 302/312 AND STUDENT TEACHING) PREPARED YOU ADEQUATELY FOR TEACHING?

STRONGLY AGREE ____
AGREE ____
NEUTRAL ____
DISAGREE ____
STRONGLY DISAGREE ____

THANK YOU FOR YOUR TIME.

THOMAS P. ROY
ASSISTANT COORDINATOR
TEACHER EDUCATION SERVICES

TP/01
ELEMENTARY EDUC

THE 9900, 16-BIT MICROCOMPUTER
IN THE
EDUCATIONAL ENVIRONMENT

Donald R. Scherer, Ph.D.
Seton Hall University

The microcomputer now provides the educator with a cost-effective alternative to timesharing. This paper discusses the Technico 9900 microcomputer and its educational uses. The Technico 9900 microcomputer was chosen because of its 16-bit, minicomputer-based, state-of-the-art architecture and because its software/hardware development is directed toward the educational market.

Use of the microcomputer is discussed from a historical perspective so that the microcomputer's current place in educational computing may be better seen. The microcomputer, for this paper, is not viewed as a total replacement for our current computing, but rather a suggestion is made to restructure the approach to beginning computer courses from higher level languages to assembly language.

Introduction

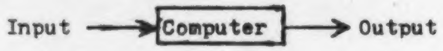
The electronic digital computer has been a reality in our world for over 25 years. From its earliest days it has been used in an educational environment, first, to educate engineers, businessmen, and programmers about itself. Then as "spare time" became available the computer was used with "in-house" educational programs in the industrial environment. As the costs decreased still further the computer's educational use extended into universities, colleges, high schools and primary schools. There has never been any doubt as to using the computer in education, but the key element has always been cost, both hardware and software.

Except for the earliest times before operating systems and timesharing when "stand alone" was the only mode of operation, it has been economically difficult to devote a computer solely to educational use. Now, the age of the microcomputer has changed the economic situation. The microcomputer has brought us back to the "beginnings," but with all the modern architecture available in many big computers and at an unbelievably low cost. The dream of placing a complete modern computer in the hands of a student is again a reality.

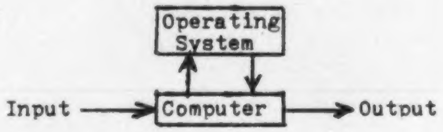
However, it is not as simple as putting your cash on the line and then business as usual. The microcomputer is a hardware revolution. At this point in time the educator must supply the software, or at least part of it. Software creation also means "software control." It means that we may keep all our programs and timesharing, yet gain primary computer control for ourselves and our students with a minimum of additional cost. Even our present computer terminals are compatible with the microcomputers!

COMPUTER CONFIGURATIONS

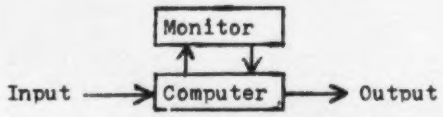
1950's Big Computers



1960's Bigger Computers



1970's Micro/Mini Computers



Historically, we may briefly look at computers from an access configuration (see illustration). The 1950's had stand-alone computers, that is, when you had access to the computer, you had all the computer facilities available. The 1960's changed that with the operating system which allowed multiple users. While the minicomputers emulated (as best they could) their bigger cousins, the microcomputers have emulated the well developed minicomputers. The 1970's see again the stand alone computers, but with a monitor system based on what we have come to expect from our timesharing experience. The 1970's really contain the full gamit of the past, and we as educators must be careful that we do not lose the advantages gained over 25 years when we contact the microcomputers. The point is that many microcomputers will drop you off in the 1950's unless you look carefully at a monitor/operating system which meets your needs.

Microcomputer Terminology

An educational microcomputer for this paper is a full computer with all its computing functions accessible by means of a standard computer terminal. We are not referring here to hardware teaching devices as being microcomputers, although microcomputers may serve in that capacity, nor are devices which use only numeric input and output of data to be considered microcomputers, even though they use a microprocessor as their computing element.

An educational microcomputer to be properly so-called should include a minimum Monitor and Assembler. While it must use a microprocessor as its computing element, it should also incorporate some PROM (Programmable Read Only Memory) in which to store its Monitor and Assembler. This is preferable, since if these are stored in RAM (Random Access Memory), they must be loaded each time the system is turned on. This is difficult in an educational environment, and increases the cost of the microcomputer since additional equipment such as cassette tape recorder or floppy disc would be needed to store the Monitor and Assembler. Not having the Monitor and Assembler in PROM also increases the dificulty in "bringing up the system" and may well preclude the use of the microcomputer in less technical environments such as the primary grades.

The educational microcomputer should contain some static or dynamic RAM for the user, but since most programs used in teaching are small, the RAM size need not be large. Two to four "K" of RAM should be more than sufficient for most Assembly language programs; this assumes that the Monitoer and Assembler are stored in PROM. Finally, the microcomputer should contain an I/O (Input/Output) port for a full alphanumeric keyboard printing or CRT (Cathode Ray Tube) /Video Terminal.

The educational microcomputer need not include a cassette tape recorder or floppy disc drive. The cassette tape recorder is a very handy and low cost addition to the microcomputer, however, a disc drive, floppy or otherwise, is a much higher price addition, five times or better the cost of the entire microcomputer! Unfortunately the disc drive also adds considerably to the cost of additional memory (RAM) needed to support the disc. While the addition of disc drives bring the microcomputer into the realm of a well-equipped minicomputer for a half or less the cost of the mini, it also brings the cost of the microcomputer we have been talking about up about ten times! To put this in terms of hard cash, the basic microcomputer with I/O port, 2 K bytes of RAM, Monitor, Assembler, and power supply costs approximately $500.00, whereas the microcomputer plus disc and higher level language software costs approximately $5000.00. A ten-to-one reason for seriously considering the teaching of beginning computer science using Assembly language.

The Technico 9900 Capabilities

The Technico 9900 microcomputer fits all the above criteria for an educational microcomputer. From the user's viewpoint it supports a full ASCII keyboard input and printer/CRT (20 ma. loop, TTL, or EIA) output, with 50 to 9600 baud speed selectable by merely typing the letter "X" after pressing the RESET button. This provides for easy terminal interchange and acceptance of a wide variety of available terminals.

The Technico 9900 microcomputer has a Monitor type operating system incorporated into its PROM so that it is always available and will not be accidentally erased. The Monitor in addition to the selectable baud rate mentioned above contains eleven general purpose commands which allow the user (with a minimum of verbiage) to do such things as: execute programs, edit existing programs, display the contents of memory and registers, put breakpoints into existing programs, take snapshots of memory and registers at a given point in a program, and also do hexadecimal arithmetic. One final bonus of the Monitor is that it is set up so that the user may incorporate into his own programs many of its I/O and other routines with a single instruction, thus saving programming time and memory space.

The Technico 9900 microcomputer also has available an "on-board" Assembler in its PROM. The Assembler is unique and especially valuable for educational use in that it is "interactive." What happens is that each assembly language instruction is immediately converted to machine language so that the student may readily see the one-to-one conversion occur, literally in fornt of his eyes! For the student, this greatly helps to establish the relationship between assembly language and machine language. An edit is also included which checks for syntax, op code, and operand errors, allowing correction and immediate re-entry of the incorrect instruction.

For the hardware minded, the Technico 9900 microcomputer will accept as a simple "plug-in", 2K bytes of EPROM (Erasable Programmable Read Only Memory), and it has an on-board EPROM programmer, so that the user may keep 2Kbytes of his own programs in a semi-permanent state. EPROM is erased using ultraviolet light and so may be re-programmed, unlike PROM or ROM which once programmed cannot be undone.

A Return to Fundamentals

In the past and present most educational computing time was taken from a batch or timeshare system in which industry or at best, educational administration, had primary control. Most computing was and is done in higher level languages. Higher level language computing makes the computer itself "transparent," that is unrecognizable to the user. For this type of operation, the ideal situation is that FORTRAN, should be FORTRAN, no matter what computer it is running on.

The point of the above is that by beginning the introduction to the computer with a higher level language, the student does not really see or work with a computer. Rather, the student works with a "language," and one which necessarily masks the computer itself! The higher level language, or better that part of it which the student uses, is really a controller of a "printer" and a mathematical entity, rather than an exploration of the many facets of a computer. Even current textbook analogies describe the computer to the student as some sort of fast and efficient secretary! This is not to say that an introduction to computer science using higher level languages is wrong, but rather this type of approach tends to get away from the very thing we are studying, namely the computer, itself.

In the past educators had little choice, it was higher level languages or nothing. In many educational situations the student never even saw the computer, for it was many miles away at the end of a telephone line. The microcomputer changes this, but to realize its low cost and I mean its $500.00 cost, it means we, as computer science educators, must look seriously at a beginning course in computer science using assembly language.

For many who were brought up on higher level languages, going back to assembly language is almost untenable. Even in industry, to change a COBOL programmer to assembly language requires the serious threat of job loss to provide the necessary incentive! However, to change an assembly language programmer to a higher level language requires no effort at all, and the option for return to assembly language at a later date is always available without great hassle. I believe the above strongly suggests that educators rethink using assembly language in beginning computer science courses.

There is one further argument for a return to assembly language now that microcomputers have made it cost-effective. As computer time has become more available the introduction of computers to high school and elementary students is progressing. For anyone who has taught computer science from the adult to elementary levels, comes the realization that the seemingly tedious programming concepts of assembly language are less of an instructional problem as one moves toward the lower grades. Yet, the very reason that computer science is not introduced at these earlier levels is the high cost of computing! Here, in the microcomputer are the answers to the problems of cost and teaching computer fundamentals.

Software and the Microcomputer

Microcomputer hardware is right up there with the best of the minicomputers, in fact, much microcomputer hardware was copied from a successful minicomputer. For example, the 9900 was taken from Texas Instrument's 990 minicomputer series; the LSI-11 was taken from Digital Equipment Corporation's PDP 11 minicomputer series. While the microcomputer hardware is state-of-the-art, its software is still catching up. Of course, in mini-derived microcomputers some of the mini's software may be run on the microcomputer, but not without a considerable upgrade in memory size and peripherals such as disc drives. If you expect the microcomputer to act like a large minicomputer, you will be disappointed.

Microcomputer software is still developing. FORTRAN and BASIC are available, but to use them you must increase memory size and add at least one floppy disc - with these additions go much of the cost advantage of the microcomputer for education. As was noted before, to put this into dollars and cents, it means increasing the cost from approximately $500.00 to $5,000.00, a ten to one increase! It might be noted that the $5,000.00 is less than a comparable minicomputer, but it is mostly hardware cost, software and maintenance must be added.

When one is talking about a one board microcomputer, such as is possible with the Technico 9900, maintenance becomes a matter of mailing the board off for repairs and replacing it with a spare whose cost is approximately $300.00. (This is not to imply that the Technico 9900 is not available with disc, FORTRAN, BASIC, etc., for it can be purchased with all these additions). However maintenance is an important consideration which may be overlooked in the excitement of the moment.

The Instant Input Assembler (IIA)

The Instant Input Assembler of the Technico 9900 microcomputer allows the student and educator to capitalize on a fantastic dollar saving and at the same time to have hands-on use of a 69 instruction set complete computer. The IIA is an excellent teaching aid as well as a memory and cost svaer. The IIA is stored in a small amount of non-erasable PROM, 1K bytes to be precise. It does not have to be loaded - a push of the RESET and it and the Monitor are ready to go.

At this writing, the Technico 9900 microcomputer is the only one to contain an IIA. This gives it a tremendous dollar advantage, for it permits full assembly language computer programming with a minimum cash outlay. For its small space it does have one disadvantage, and that is it does not allow the use of labels. However, all addressing modes are available, and one may use indexed, indirect, direct or autoincrement register addressing, plus direct, immediate and relative memory addressing.

The 9900 in Computer Science Education

At the university/college level, the 9900 allows unrestricted assembly language programming. There is no need for a simulator or fear that the use of assembly language will allow the student access to security areas of the computer's files. The addition of a cassette tape recorder will allow the students to copy their own programs and to reload them at a later date. If the university is already heavily invested in timesharing terminals, an approximately $500.00 addition to each terminal allows the 9900 microcomputer addition, which may be a very cost-effective way of increasing computer capacity without going to a larger timesharing computer.

Since the 9900 is a single board unit in its basic configuration and is packaged so that access to the integrated circuits is easily accomplished, its use in a computer hardware course is ideal. If a professor so wishes he may place LED's (Light Emitting Diodes) at various points on the board to indicate logic levels. Also the clock may be easily slowed down to use with less expensive oscilloscopes of reduced bandwidth.

For introductory microcomputer courses, the 9900 has been successfully used when introduced as a kit for the students to build. Very few assembly problems have been encountered and its kit cost for students has been very low, while offering all the advantages of being able to work with 16-bit minicomputer architecture.

The 9900 microcomputer also fits in well with data structures courses in the university/college environment. It may also have a consideration in presenting a new course in mini-business data processing. There are currently many attempts to adapt the microcomputer to a data base management system (DBMS) and an accounting system for small businesses.

Assembly language has not often been used in the high school environment, but whenever it has been taught, it has been successful and very valuable for the students going on to university/college computer science courses. It seems to foster a more concrete algorithmic way of thinking for the student, clearing the way for an easier grasp of the computer science principles being taught in the higher level courses.

It would seem the above should also extend to the primary grades. After all their reasoning power is no less than that of high school or college students, it's just that they have had fewer experiences to draw upon. The fewer experiences may also mean fewer "hang-ups" and an easy teaching assignment for the computer science taecher. My own experiences have found this to be true and an introduction to computer science at the primary level with a real computer present to work with could be a delight for the student and the teacher

The 9900 in Other Curricula

It would be hard to imagine a single board 9900 for use in science, mathematics, or arts without first demanding student knowledge of assembly language programming. At this point in time that demand is not easily met, although the future might not find that difficult at all. However, we live in the present and so cannot expect assembly language programming (or programming of any type) by the instructor. We are thus limited to canned programs for such curricula and as we know from the past, good "courseware" is hard to come by.

The microcomputer since it is mini-based will, given sufficient memory, run most of the programs now designed for minicomputers. With a disc and higher level language capability the microcomputer will be able to be used for motivational purposes and CAI (Computer Assisted Instruction), including drill and practice sessions and simulation. While the potential for the microcomputer in these areas cannot be denied, it must be realized at this point that it is still potential which has yet to be activated.

The 9900 Compared with Other Micro's

The attempt here is not to make a hardware comparison, but rather a comparison based on present educational needs. In considering a microcomputer for educational use, the following questions must be asked and answered, hopefully affirmatively:

1) What is the selected microcomputer's total cost, including power supplies, peripherals, if required, and software?

2) Is software, such as Monitors, Operating Systems, Assemblers, etc., included in the above cost, or extra?

3) What is the cost of maintenance and how is it accomplished?

4) Will the selected microcomputer meet the educational need?

5) Is the selected microcomputer software and hardware expandable to meet forseeable future needs?

6) Is it possible to establish a trial situation before going "full bore"?

7) Are educational materials available to teach the use of the selected microcomputer, or must it be gleaned from computer manuals?

8) Is educational courseware available for the selected microcomputer?

With most current microcomputers fitting to a broad range of pocketbooks, the 9900 is no exception, it ranges from a low of $299.00 in kit form to $4,000.00 assembled with 32K bytes of memory and two floppy discs. The real advantage of the 9900 lies in the fact that a Monitor and Assembler are available in an assembled configuration for approximately $600.00. To this price must be added some sort of terminal, preferably a printing terminal, since it makes the handling of assembly language (or any other language for that matter) easier to study and of course necessary for the submission of student work.

As far as hardware structure, the LSI-11 and the NOVA are 16-bit computers similar to the 9900, however, their comparable hardware costs are higher than the 9900 and they are not available in the minimum configuration of the 9900. The Z80, 8080, and 6800 based microcomputers are more for the technically inclined hobbiest, rather than the educator.

The Technico 9900 does have a software advantage over other microcomputers in that it has an on-board Monitor and Assembler. The Assembler is perhaps the most important, since for other systems an assembler requires 8K bytes of memory and for some a bit more. The LSI-11, the most costly microcomputer, does have the widest variety of software, but it must be purchased individually.

Maintenance is the most hidden cost and the hardest to really get at. Nothing beats on-site maintenance, but for microcomputers this is not routine even if you live down the street from the manufacturer. Even assembled microcomputers require you to plug the components together and determining which is bad if you have a sizeable system may not be easily accomplished unless you have some flair for trouble shooting. However, if you stay with the single board Technico 9900, they have, at this writing, a $25.00 fee for a board checkout, which is a better price than most video board manufacturers will give for a less complicated board.

Determination of meeting an educational need will be left for another paper, but that question did prompt question 6). A trial with a smaller amount of equipment may sometimes answer question 4) when the equipment is new (as with microcomputers) and other field trials have not been made. It might be mentioned that this opens the possibility for some research funds for educational uses of microcomputers.

The final questions 7) and 8) turn up few educational materials. Heathkit has generally good educational matter, but the microcomputer materials have not been seen by this author. Technico is in the process of publishing a self-instructional handbook on the 9900 which should be a winner for easing into its monitor and assembly language.

In summary, the microcomputer is new, the educator would best review the eight questions and spend cautiously.

COMPUTER STUDIES IN THE JUNIOR HIGH SCHOOL: A THREE-YEAR CURRICULUM SEQUENCE

HOWARD WEINMAN
HALSEY J.H.S.
REGO PARK, N.Y.

ABSTRACT: A Three Year Computer Studies Curriculum Sequence (TYCSCS), for bright junior high school students, will be discussed. The following questions will be answered: (1) Why do so few junior high schools offer any computer studies courses? (2) Why should this course of study be undertaken at such an early age? (3) How do we choose pupils for our program? (4) What are the major objectives of the TYCSCS? (5) What equipment do we use to implement our program? (6) What topics are included in the syllabus? (7) In what sequence should the subject matter be presented? (8) How much time should be allocated to each topic? (9) How should these topics be taught?

INTRODUCTION

We are well into the second half of the twentieth century and the computer revolution is an unending reality. Everyone should be educated accordingly. The computer revolution has been accompanied by an information explosion. As a result, there is so much more to be learned today. Major revisions in educational curricula and priorities are required to prepare our youngsters properly.

There was a time when arithmetic was taught in the university. It has since filtered down into the elementary school. Now the same thing must happen to computer-related subject matter.

In teaching junior high school students for several years, I have noticed that many of them are much more capable than most people realize. Society's underestimation, of our youngster's abilities, is one of the major impedments to the implementation of computer studies curricula in the early grades. Other obstacles include: (1) a lack of equipment and money to obtain it (2) a shortage of knowledgeable instructors, (3) a scarcity of time and money for curriculum development, (4) the generally conservative attitude of educational administrators.

The equipment problem will be minimized by the development of inexpensive microcomputers. A new breed of recent college graduates (or soon to be) will fill the instructor gap. The success and exposure, of several independently implemented programs, will help solve the curricular and administrative resistance difficulties. One such program is the Three Year Computer Studies Curriculum Sequence (TYCSCS), which is the topic of this paper.

STUDENTS

Students are selected for our TYCSCS program based on the following criteria:
(1) city-wide math test and reading test scores,
(2) math & science report card marks,
(3) sixth-grade teacher's evaluation,
(4) students' own choice.

The regular math course for these students is the Secondary School Math Curriculum Improvement Study (SSMCIS) program. The unified approach used in SSMCIS is excellent and I highly recommend it.

OBJECTIVES

The students will be the leading technologists of tomorrow. The TYCSCS program seeks to prepare them for this role. The specific goals of TYCSCS are many and varied. At the conclusion of the three-year course of study:

(1) 100% of the students will have significantly increased their understanding of the computer's role in our society as measured by teacher developed pre-test and post-tests.

(2) 90% of the pupils will have an in-depth knowledge of computer programming as demonstrated by the successful completion of independent programming projects.

(3) 75% of the students will have attained sufficient programming expertise to be able to employ it as a tool to enhance education in other subject areas. This achievement will be demonstrated by the successful completion of appropriate programs.

(4) 90% of the pupils will have reinforced and extended the knowledge gained in the math classroom as evidenced by their performance on teacher developed tests and teacher assigned problems.

(5) 90% of the youngsters will have improved their abilities to function independently as demonstrated by the successful completion of individualized programming projects.

(6) 100% of the pupils will have an improved understanding of the various types of job opportunities created by the computer revolution. This will be measured by teacher developed tests.

EQUIPMENT

Presently we are trying to accomplish our objectives with the aid of the following equipment:

(1) 6 COMPUCORP 025E Programmable Calculators ($495 each). Each has a 2½" tape for numerical printout. The 025E is programmable only in its own language. Programs may be loaded via the keyboard or a card reader. Data is entered through the keyboard. Each 025E contains 256 steps of memory and 10 storage registers.

(2) 1 WANG 2200 PCS ($5200). The 2200 is programmable only in an extended BASIC. The alphanumeric keyboard is the primary input device while the CRT is used for output. Our 2200 has 8K bytes of core and employs cassette tapes for auxiliary storage.

(3) 1 DIGIAC 2010 LOGIC TRAINER ($325). The 2010 contains pre-wired AND, OR, NOT, NOR and NAND gates as well as flip-flops. It also includes a power supply, clock, input switches and output lamps. The machine is quite useful in verifying truth tables and the simplification of Boolean expressions. It is also helpful in designing computer circuitry. The 'hands-on learning' effect achieved by the logic trainer is invaluable in reinforcing the abstract chalkboard theory that is taught to the pupils.

SUBJECT MATTER

The equipment described above is employed in order to maximize the students' absorption of the TYCSCS subject matter which is outlined below in TABLE #1. The symbols in parentheses indicate where the corresponding information can be found in TABLE #2.

TYCSCS TOPICAL OUTLINE (TABLE #1)

I. COMPUTER LITERACY (IA)
  A. PRESENT & FUTURE COMPUTER APPLICATIONS
  B. HISTORY OF COMPUTATIONAL DEVICES
  C. COMPUTER SYSTEMS & COMPONENTS
  D. JOB OPPORTUNITIES
II. COMPUTER MATHEMATICS
  A. BINARY, OCTAL & HEXADECIMAL (IF1)
  B. RANDOM NUMBERS (IF6, IH9)
  C. BOOLEAN ALGEBRA & LOGICAL DESIGN
    1. BASIC GATES (IIH1)
    2. DE MORGAN'S LAWS (IIN2)
    3. SIMPLIFICATION OF BOOLEAN EXPRESSIONS (IIN4)
    4. FLIP-FLOPS (IIIF1)
    5. COMPUTER CIRCUITS (IIN5,IIIF, IIIG4E)
III. COMPUTER PROGRAMMING
  A. CONCEPTS & TECHNIQUES
    1. FLOWCHARTING (IE)
    2. LOOPING, COUNTING (IF2,IH2,IH4)
    3. DECISION MAKING (IF2, IH3)
    4. SUBPROGRAMS
      a. FUNCTIONS (IIJ1)
      b. SUBROUTINES (IF5,IH6,IIJ3)
    5. NESTED LOOPS (IF8,IID1b)
    6. NESTED SUBROUTINES
    7. DATA STRUCTURES
      a. LISTS (IID1a)
      b. TABLES (IID2b)
    8. STRING MANIPULATION (IIJ2)
    9. PROGRAM CHAINING (IIID2)
    10. REGISTER SPLITTING (IIIB1)
    11. DEBUGGING (IF3, IID3)
    12. DOCUMENTATION (ID,IF,IH3)
  B. LANGUAGES
    1. 025E MACHINE LANGUAGE
      a. STR, RCR (IB4, ID)
      b. INT/FR, 2ND/FUNC (IB5, IF2)
      c. ENEO, THEN, JUMP, EGTO (IF2)
      d. BR, RCLP (IF5)
      e. SPECIAL FUNCTION 4 (IF7)
      f. SPECIAL FUNCTION 6 (IF4)
      g. SFI4, SKF4, RFI4 (IIB2)
      h. OTHER ADVANCED INSTRUCTIONS (IIIG4a)
    2. BASIC
      a. STATEMENTS
        i. PRINT, LET, END, STOP (IH)
        ii. INPUT, GOTO, IF-THEN (IH)
        iii. REM, FOR, NEXT (IH)
        iv. READ, DATA, RESTORE (IH)
        v. GOSUB, RETURN, ON--GOTO (IH)
        vi. DIM, KEYIN (IID)
        vii. DEFEN, GOSUB', DEFFN' (IIJ)
        viii. PRINTUSINB, IMAGE, INIT (IIJ)
        ix. MAT OPERATIONS (IIID1)
        x. COM, LOAD, COM CLEAR (IIID2)
        xi. CONVERT, RETURN CLEAR (IIID3)
        xii. DATALOAD, DATASAVE (IIID4)
        xiii. SELECT (I/O DEVICE) (IIID5)
        xiv. OTHER ADVANCED STATEMENTS (IIG4b)
      b. FUNCTIONS
        i. SQR, INT, ABS (IH)
        ii. RND, TAB, SGN (IH)
        iii. HEX (IID2)
        iv. STR, LEN, POS, NUM (IIJ2)
    3. FORTRAN (IIIG4c)
    4. COBOL (IIIG4d)
IV. MACHINE OPERATION
  A. COMPUCORP 025E (IB, ID, IF)
  B. WANG 2200 PCS (IG, IH)
  C. DIGIAC 2010 LT (IIH)
V. PROBABILITY & STATISTICS
  A. MEASURES OF CENTRAL TENDENCY (IJ1)
  B. MEASURES OF DISPERSION (IJ2)
  C. PROBABILITY RATIO & RELATIVE

FREQUENCY (IJ3)
- D. NORMAL DISTRIBUTION (IIF1)
- E. CORRELATION & REGRESSION (IIL1)
- F. HYPOTHESIS TESTING (IIIG4f)
- G. STUDENT'S T-DISTRIBUTION (IIIG4f)
- H. CHI-SQUARE DISTRIBUTION (IIIG4f)
- I. CURVE FITTING (IIIG4f)

INSTRUCTIONAL SEQUENCE

The sequence, in which the subject matter is presented, is illustrated in TABLE #2. The numbers in parentheses indicate the numbers of days devoted to the corresponding topics.

TYCSCS TEACHING OUTLINE (TABLE #2)

- I. THE FIRST YEAR
  - A. COMPUTER LITERACY LECTURES (15 days)
    - 1. PRESENT AND FUTURE COMPUTER APPLICATIONS
    - 2. HISTORY OF COMPUTATIONAL DEVICES
    - 3. COMPUTER SYSTEMS & COMPONENTS
    - 4. JOB OPPORTUNITIES
  - B. COMPUCORP 025E AS A CALCULATOR (8)
    - 1. GET FAMILIAR WITH MACHINE
    - 2. 4 BASIC BINARY OPERATIONS & SQ.RT.
    - 3. PRINTING
    - 4. STORING & RECALLING
    - 5. INT/FR, 2ND/FUNC
  - C. DATA TRANSFER (3)
    - 1. REGISTER LEVEL DATA MOVEMENT
  - D. KEYBOARD PROGRAMMING (10)
    - 1. STRAIGHT LINE PROGRAMS
      - a. WRITING
      - b. LOADING
      - c. EXECUTING
  - E. FLOWCHARTING (6)
    - 1. DESIGN & ANALYSIS
      - a. FAMILIAR SITUATIONS
      - b. MATHEMATICAL PROBLEMS
      - c. GAMES
      - d. PRACTICAL APPLICATIONS
  - F. CARD PROGRAMMING (36)
    - 1. CODES, CARDS & MEMORY (3)
    - 2. LOOPING & DECISION MAKING (15)
    - 3. DEBUGGING (3)
      - a. STEPPING THROUGH PRINTOUT & PROGRAM SIMULTANEOUSLY
    - 4. IDENTIFIERS (2)
    - 5. SUBROUTINES (3)
    - 6. RANDOM NUMBERS (4)
    - 7. SENSE KEY (3)
    - 8. NESTED LOOPS (3)
  - G. INDIVIDUAL PROJECT WORK (18)
    - 1. 025E PROGRAMS
    - 2. OPERATION OF WANG 2200 PCS
      - a. ROTATING GROUPS
      - b. DEMO GAME PROGRAMS
  - H. BASIC PROGRAMMING (20)
    - 1. PRINT, LET, END (2)
    - 2. INPUT, GOTO, STOP (5)
    - 3. REM, IF--THEN (2)
    - 4. FOR & NEXT (5)
    - 5. READ & DATA & RESTORE
    - 6. GOSUB & RETURN, SQR, INT, ABS (3)
    - 7. ON--GOTO
    - 8. TAB (1)
    - 9. RND (2)
  - I. INDIVIDUAL PROJECT WORK (18)
    - 1. 025E PROGRAMS
    - 2. BASIC PROGRAMS
  - J. PROBABILITY & STATISTICS (7)
    - 1. MEASURES OF CENTRAL TENDENCY (2)
      - a. MEAN
      - b. MEDIAN
      - c. MODE
    - 2. MEASURES OF DISPERSION (3)
      - a. RANGE
      - b. VARIANCE
      - c. STANDARD DEVIATION
    - 3. PROBABILITY RATIO & RELATIVE FREQUENCY (2)
  - K. INDIVIDUAL PROJECT WORK (19)
- II. THE SECOND YEAR
  - A. INDIVIDUAL PROJECT WORK (10)
    - 1. REVIEW WORKSHEETS ON 025E & BASIC
    - 2. 025E PROGRAMS
    - 3. BASIC PROGRAMS
  - B. 025E PROGRAMMING (6)
    - 1. INTERNAL DATA FORMAT (2)
    - 2. FLAGS (2)
    - 3. ROMS (2)
  - C. INDIVIDUAL PROJECT WORK (10)
  - D. BASIC PROGRAMMING (11)
    - 1. DIM
      - a. LISTS (2)
      - b. TABLES (3)
      - c. MEMORY ALLOCATION (1)
    - 2. HEX CODES (3)
      - a. CHARACTER REPRESENTATION
      - b. CURSOR CONTROL
    - 3. DEBUGGING (2)
      - a. TRACE MODE
      - b. INSERTION OF PRINT & STOP
    - 4. KEYIN
  - E. INDIVIDUAL PROJECT WORK (12)
  - F. STATISTICS (7)
    - 1. NORMAL DISTRIBUTION (5)
    - 2. GRAPHING LINEAR EQUATIONS (2)
  - G. INDIVIDUAL PROJECT WORK (7)
  - H. LOGIC (12)
    - 1. TRUTH TABLES, VENN DIAGRAMS, LOGIC GATES, ELECTRICAL CIRCUITS
      - a. AND, OR, NOT (4)
      - b. A→B, A↔B, A⊕B (4)
      - c. MODUS PONENS, SYLLOGISM (4)
  - I. INDIVIDUAL PROJECT WORK (12)
    - 1. 025E PROGRAMS
    - 2. BASIC PROGRAMS
    - 3. LOGIC
  - J. BASIC PROGRAMMING (11)
    - 1. USER-DEFINED FUNCTIONS (DEFFN)(1)
    - 2. STRING MANIPULATION (5)
      - a. STR, LEN, POS, NUM, INIT
    - 3. MARKED SUBROUTINES (GOSUB' & DEFFN') (3)
    - 4. PRINTUSING & IMAGE (2)
  - K. INDIVIDUAL PROJECT WORK (12)
  - L. STATISTICS (8)
    - 1. CORRELATION & REGRESSION
  - M. INDIVIDUAL PROJECT WORK (8)
  - N. LOGIC (17)
    - 1. NAND, NOR (4)
    - 2. DE MORGAN'S LAWS (4)
    - 3. NAND GATE IMPLEMENTATION (4)
    - 4. SIMPLIFICATION OF BOOLEAN EXPRESSIONS (3)
    - 5. EQUALITY DETECTOR (2)
  - O. INDIVIDUAL PROJECT WORK

III. THE THIRD YEAR

- A. INDIVIDUAL PROJECT WORK (15)
  - 1. REVIEW WORKSHEETS ON 025E, BASIC, LOGIC & STATISTICS
  - 2. 025E PROGRAMS
  - 3. BASIC PROGRAMS
  - 4. LOGICAL DESIGN
- B. 025E PROGRAMMING (2)
  - 1. SPLITTING REGISTERS
- C. INDIVIDUAL PROJECT WORK (8)
- D. BASIC PROGRAMMING (9)
  - 1. MAT OPERATIONS (2)
  - 2. COM, LOAD, COM CLEAR (2)
  - 3. CONVERT, RETURN CLEAR (2)
  - 4. DATASAVE, DATALOAD (2)
  - 5. SELECT (I/O DEVICE) (1)
- E. INDIVIDUAL PROJECT WORK (15)
- F. LOGICAL DESIGN (16)
  - 1. FLIP-FLOPS (9)
    - a. R-S
    - b. CLOCKED R-S
    - c. J-K
  - 2. BINARY UP COUNTER (3)
  - 3. SHIFT REGISTER (3)
  - 4. HALF-ADDER (3)
- G. INDIVIDUAL PROJECT WORK (85)
  - 1. 025E PROGRAMS
  - 2. BASIC PROGRAMS
  - 3. LOGICAL DESIGN
  - 4. OPTIONAL INDEPENDENT STUDY TOPICS
    - a. 025E PROGRAMMING
      - i. INDEX REGISTER INSTRUCTIONS
      - ii. SHIFT INSTRUCTIONS
    - b. BASIC PROGRAMMING
      - i. BIN, VAL, BOOL
      - ii. AND, OR, XOR
      - iii. ADD, ROTATE
      - iv. HEXPRINT
      - v. PACK, UNPACK
      - vi. ON ERROR GOTO
      - vii. IF END THEN
    - c. FORTRAN
    - d. COBOL
    - e. LOGICAL DESIGN
      - i. PARITY BIT GENERATOR
      - ii. KARNAUGH MAPS
      - iii. FULL ADDER, PARALLER ADDER
      - iv. SUBTRACTOR, MULTIPLIER
      - v. ASSORTED COUNTERS
      - vi. BINARY-DECIMAL CONVERTER
      - vii. DECIMAL-BINARY CONVERTER
      - viii. OTHER CODE CONVERSIONS
    - f. STATISTICS
      - i. HYPOTHESIS TESTING
      - ii. STUDENT'S T-DISTRIBUTION
      - iii. CHI-SQUARE DISTRIBUTION
      - iv. CURVE FITTING

INSTRUCTIONAL STRATEGY

The COMPUTER LITERACY lectures are distributed throughout the course. Slides, films, transparencies, magazine articles and cartoons are all employed to enhance this segment of the syllabus.

The students must be allowed and encouraged to use the machines as soon and as much as possible. Therefore, almost immediately pupils begin using the 025Es as calculators. During this time students familiarize themselves with the operation of the machines via "HANDS-ON" learning; one of the major features of the course.

As we proceed through the curriculum a progressively larger number of class-lecture type lessons are replaced by individual project work. During the early part of the course the youngsters are taught the fundamentals of 025E and BASIC programming. After each day of lecture, the students are given time to go to the machines in groups of 3 or 4 to reinforce the lecture material via HANDS-ON learning. (The time allocation indicated in TABLE #2 includes both the number of sessions for lecture as well as group work.) Generally the group work allows the pupils to complete a problem solution begun in the previous day's lecture (i.e., finish writing, loading, and executing a program). Gradually the pupils are required to do more and more of the work on their own.

Each class-oriented segment of TYCSCS is followed by a number of days of INDIVIDUAL PROJECT WORK. Each pupil chooses the particular projects on which he wishes to work. In some cases, the teacher specifies the general topic from which the youngster may choose (e.g. a particular unit in SSMCIS-possibly modular arithmetic). At other times the student is free to make a selection without any restrictions (e.g. Prime Number Generator, Balancing Chemical Equations, Simulation of Craps or Football, Calculation of Biorhythms, Computer Assisted Practice in (Zn,+) or Design of a Binary to Decimal Converter). While working individually students develop and improve many skills including logical thinking and report writing. Computer programming becomes a powerful, versatile tool.

Considering the class size (approximately 30) as compared to the number of machines and 1 teacher, it is imperative that the equipment and instructor be accessible to the pupils beyond just the regularly scheduled class period. As a result, both are available to the students before school, during lunch and after school.

Presently, there is no textbook for the course. Instead students are given rexographed sheets of several types. (1) Information sheets, (2) Drill sheets, and (3) Summary Exercise sheets are all used to supplement the lectures. In addition, there is a wide variety of reference books and magazines available to students in class and for home use. Included are math and science texts, computer programming texts and reference manuals, game and puzzle books, computer hobbyist magazines, and the teacher's notes.

EXTRA

For those youngsters who find themselves totally addicted to computers, there is an extra-curricular computer club. The club delves further into computer-related subject matter. The club members visit other computer facilities either as spectators or users. Several businesses and universities have cooperated with us on this venture.

CONCLUSION

A course like TYCSCS, Three-Year Computer Studies Curriculum Sequence, is an essential ingredient in TODAY'S education of bright youngsters. These students are capable and enthusiastic. We must not hold them back. They deserve a 1978 education in which they learn computer-related subject matter in a HANDS-ON, INDIVIDUALIZED WORK environment.

Multi-usage Approaches to Computer-extended Instruction

Larry L. Hatfield
Associate Professor
Department of Mathematics Education
105 Aderhold Hall
University of Georgia
Athens, GA 30602

A rationale for a more mature application of computers to instruction is presented. A planning model for a comprehensive use of the computer in instruction is described. A multiplicity of types of computer experiences is considered in relation to the objectives or topics of the curriculum. The computer uses featured include student programming, practicing, simulating, gaming, tutoring, and testing. Particular multi-usage instructional packages under development for school mathematics topics are discussed to illustrate the use of the planning model. Extensions of the application of a multi-usage perspective are considered.

Throughout the world educators are exploring the potential of the computer in instruction. These efforts encompass a wide range of educational levels, spanning pre-school through graduate school. Typically the computer has been wisely viewed as a tool for enhancing or extending the learning of the subject matter knowledges to be taught in the school's curriculum. Applications of computers to instruction have become quite varied, including: (a) student-generated programming, (b) practicing, (c) simulating (d) gaming, (e) tutoring, and (f) testing.

The computer has proven to be a provocative, motivating device in most ways it has been used. Students and teachers alike seem to become excited and enthusiastic about the computer as an extension of the classroom. Yet in their eagerness to incorporate such stimulation into instruction many educators have failed to reflect upon the ways and places in a curriculum where the computer might indeed promote the achievement of the chosen goals and objectives.

It is typically the case that any computer application to instruction reflects a single strategy for using the machine. That is, it appears that the conceptualization of the specific computer use has often grown out of a question such as the following:

> Knowing how to employ computer usage x, what can we find in the school curriculum that will permit us to employ x?

This attitude is exhibited by the computer enthusiasts who promote a single, particular type of computer application to instruction, such as simulation. For example, the Huntington Project identified numerous choice topics from the school sciences and mathematics for which clever simulation programs could be written. Similarly, much of the early development of tutorial and practice CAI involved prototypic computer programs for a variety of isolated topics, often chosen to portray the potential of CAI. The obvious difficulty with this approach to applying the computer to instruction is the potential distortion to the school curriculum.

To be sure, a similar, more mature question appears to have served as basic to some computer applications:

> After planning and/or reviewing the school curriculum we wish to have, how can we employ computer usage x to facilitate our instruction?

According to the intent of this guiding question, the more comprehensive planning of an overall curriculum precedes the development and implementation of the particular use of the computer. However, the computer utilization still involves only strategy, such as student programming or practicing or testing. Various projects appear to have adopted this approach, such as the University of Denver Computing and Mathematics Curriculum, the Stanford University arithmetic drill and practice curriculum, the Scott, Foresman Computer Assisted Mathematics Program, and the tutorial courses for community colleges developed for TICCIT and

433

PLATO.

It is a fundamental maxim for any instructional theory that any chosen teaching aid or tool must fit into carefully made decisions of the learning objectives, content sample and instructional sequence. In planning effective instruction one begins with an overall selection of the purposes (objectives and goals) for teaching the material, the particular ideas and experiences to be developed with the learners, and the strategies to be used to facilitate the learning. Eventually the teacher or curriculum planner considers the range of instructional aids to be used. Of course, the computer can be one such teaching tool. But the following more comprehensive question has not been routinely asked (even by computer enthusiasts):

> Of all the known ways that a computer can be used in instruction, which uses may be effective for the learning experiences that we wish to provide in teaching a particular topic y?

The perspective for instructional planning provided by this question results quite naturally in a multiple-usage of the computer. An idealized curriculum which incorporates multi-usage computer-extended instruction would include a variety of student and teacher contacts with the computer within each topic or instructional unit. Students might:

(a) write and execute programs to solve assigned problems;

(b) complete several computer-based practicing sessions aimed at building a reasonably automatic recall and use of certain taught ideas or procedures;

(c) observe the execution and discuss the computer output of various simulation programs which feature processes from the topics being studied;

(d) participate in computer-based gaming activities related to the ideas and problems of the unit;

(e) use a variety of computerized tutorial sessions which either introduce or reteach major ideas from the curriculum; or

(f) complete computer-based or -generated testing over the learning objectives of the unit.

It should be emphasized that these computer-related learning experiences are viewed as an extension but not replacement of the regular teacher and classroom.

One way to represent this multi-usage viewpoint employs a grid (see Figure 1) wherein the instructional content (topics or objectives) is identified along one dimension and types of potential computer uses are listed along the other dimension. A careful, creative analysis of the content to be taught would suggest settings where particular uses of the computer might enhance student learning.

Contents of Instruction
(Topics or Objectives)

Computer Uses for Instruction	A	B	C	D	E	F
Programming						
Tutoring						
Practicing						
Simulating						
Gaming						
Testing						

Figure 1. A Simple Planning Grid for Multi-usage Applications of Computers to Instruction.

Of course the eventual computer usages which might become actualized in the instructional package will depend on such real factors as the curriculum planner's knowledge and creativity, the available time to locate or develop the courseware programs to be featured, the restrictions of the students' schedules to utilize the courseware, the limitations of the computer facility, and the appropriateness of the content for various computer applications. If the computer as an educational tool represents an extension of the regular classroom instruction, then the multi-usage matrix for a particular instructional unit will likely contain many "empty" cells. Such "empty" cells indicate an absence or inappropriateness of computer uses for particular instructional objectives. Extended planning of computer applications to instruction of a particular content area may result in a more completely filled matrix, including options within cells. Eventually a teacher may have many choices from which to select in planning learning activities for a particular instructional unit. Thus, the comprehensive planning for computer uses suggested by the multi-usage matrix is open-ended. At the same time it seeks to coordinate available computer-based materials into a framework centered upon the school's curriculum. Conceiving of the computer as a multi-faceted tool for instruction relative to well chosen objectives and subject matter offers a rationale more likely to become acceptable to educators who may be apprehensive and uncertain about the role of the computer in instruction.

Examples of Multi-usage CEI

The development of several multi-usage packages has been initiated by faculty and students of the Department of Mathematics Education at the University of Georgia. Working primarily within the framework of a beginning course in computer applications to mathematics instruction, the participating preservice and inservice teachers have planned and produced various computer programs and student materials. The topic areas, chosen from school mathematics, include:

(a) measurement and estimation (appropriate to upper elementary school);

(b) slope of a line (appropriate to beginning algebra);

(c) elementary functions (appropriate to beginning algebra);

(d) consumer arithmetic (appropriate to high school general mathematics);

(e) descriptive statistics and probability (appropriate to middle school).

As an example, consider the topical area slope of a line as it might be treated in an elementary algebra course. The concept of slope is developed informally ("rise over the run") and formally ($m = (y_1-y_2)/(x_1-x_2)$). Thereafter, students will practice finding slopes of lines, solve problems (algebraic as well as numerical) involving the slope concept, and discover and apply slope relationships for parallel and perpendicular lines. In considering a multi-usage CEI approach we identified the following applications:

1) Students are assigned the programming problem of writing and testing an algorithm for computing a slope given any two points.

2) A brief tutorial program, SLOPE, is available for students to have the definition retaught whereupon several examples are generated to test comprehension.

3) A practice program, SLOPE1, offers timed drills in computing slopes from two points.

4) Students work to write and test a general program for finding any missing value in the slope equation.

5) A computer demonstration simulating the effect of varying the slope of an inclined plane upon the descent acceleration of a sliding mass is given.

6) Another practice program, SLOPE2, offers drill on finding a missing coordinate given one coordinate, the slope, and a point.

7) A tutorial program, LINES, leads the student through a guided inquiry for the generalizations that slopes of parallel lines are equal and slopes of perpendicular lines are negative reciprocals.

8) Students write and run their own programs to test whether particular sets of points are contained in parallel or perpendicular lines.

9) A gaming routine, BATTLE, permits two players to play a version of "Battleship" where slopes of the opponent's ship 'segments' are known.

10) A computer-based test allows individualized pacing and multiple-trial items.

These programs and student materials have been written and used. However, no formal evaluation or studies of effectiveness have been undertaken. There are numerous additional multi-usage applications of the computer to be attempted for this topic. The packages which exist for the other topical areas identified above similarly await further study and extension. It seems clear to us that mathematics educators, knowledgeable about mathematics teaching and the potential of multiple usages of instructional computing, can identify a rich array of computer applications to most curriculum topical areas.